NEW RESULTS IN FACTORY PHYSICS

– *INSIGHTS FROM THE UNDERLYING STRUCTURES OF MANUFACTURING SYSTEMS*

A Dissertation
Presented to
The Academic Faculty

by

Kan Wu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
December, 2009

NEW RESULTS IN FACTORY PHYSICS

– *INSIGHTS FROM THE UNDERLYING STRUCTURES OF MANUFACTURING SYSTEMS*

Approved by:

Dr. Leon F. McGinnis, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Hayriye Ayhan
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Bert Zwart, Co-Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Mark Ferguson
College of Management
Georgia Institute of Technology Dr.

Dr. Craig A. Tovey
School of Industrial and Systems Engineering
Georgia Institute of Technology

Antonius Dieker
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: July 2nd, 2009

To Yuan and Season, the source of my joy,
And to my impatient wife,
for her great patience on this journey.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Leon McGinnis, for his support when I fail, his guidance when I am lost and his understanding when I get him in trouble. From him, I learn the paradigm of a scholar with integrity, persistence and generosity. I would like to express my deep appreciation to my co-advisor Dr. Bert Zwart for his clarification on my questions and comments on this thesis.

I would also like to thank Dr. Hayriye Ayhan for her generous help and Dr. Craig Tovey for his unreserved encouragement on this endeavor. Their inspiring words gave me strength to move forward when I lost myself along the way. I am also indebted to Dr. Keung Hui. He discovered my potential and encouraged me back to school several years ago. Until now, he still keeps investigating challenging problems in manufacturing systems with me.

Many thanks to Dr. Mark Ferguson and Dr. Ton Dieker for their insightful questions in the defense, to Dr. Yan Wang for his suggestion on the failure of central limit theorem, and also to Nikhil Kumar, Abhinav Dalal and Edward Huang for their support in simulation coding.

After working in the industry for almost ten years, it is not a simple task to complete this challenging journey. This thesis could not be completed without anyone of the above and many others. Here I send my deepest appreciation to all of them.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

SUMMARY

The objective of this dissertation is to enhance the overall understanding of practical manufacturing systems by using rigorous analytical approaches, primarily queueing theory, for estimating cycle times. The scope spans from a single manufacturing process to a manufacturing system.

Performance of Single Machine Systems

Queueing theory is commonly used to evaluate the performance of manufacturing systems. However, actual applications of queueing models to single machine systems encounter some practical issues. A real machine is subject to different kinds of interruptions, such as breakdowns, setups and routine maintenance. By systematically classifying different kinds of interruptions, suitable queueing models based on the structure of interruptions are proposed, and a unified model is derived to integrate all types of interruption events. New insights are derived on the application of queueing theory to a single machine system through detailed analysis of different types of interruptions.

From a practical prospective, better models are needed for parallel batch processing in manufacturing systems. The existing G/G/1 based approximate model decomposes the cycle time into three parts: wait-to-batch times, waiting times and service times. However, through detailed examination of the model, a more accurate approximate model is proposed by examining the independence assumption between wait-to-batch

times and waiting times. The new model gives considerable improvement over existing approaches.

Performance of Manufacturing Systems

The behavior of manufacturing systems is explored by first investigating the underlying structure of tandem queues. Due to the non-renewal departure process, this challenging problem is usually tackled by the parametric-decomposition method or by diffusion approximations. To search for a better model, the behavioral structure of tandem queues has been investigated. The identified structures, termed the intrinsic gap and intrinsic ratio, come with very nice properties, which we call the heavy-traffic and nearly-linear relationship properties. Based on those properties, new approximate models are developed for two single-server stations in series. Then the model is extended to approximate the queueing time of many single-server stations in series. Based on extensive simulation results, the new models considerably outperform earlier methods based on the parametric-decomposition and diffusion approximations. By using the identified structure, a way to analyze the intricate dependence among tandem queues has been proposed.

By extending the results from tandem queues, an approximate model for an entire manufacturing system is derived. The model is used to quantify the performance of manufacturing systems and has been evaluated using complicated simulation models from an industrial partner. The results demonstrate that the new method has strong capability to describe the performance of practical manufacturing systems precisely.

CHAPTER 1

INTRODUCTION

To see a world in a grain of sand, And a heaven in a wild flower. ~ William Blake

Human beings search for productivity improvement in order to make a better life. In the mid and late-twentieth century, many innovative production methods were invented. The emergence of MRP (material requirements planning) in the 1960s, and the prevalence of JIT (just-in-time) and TQM (total quality management) in the 1980s are examples of the search for new methods to improve factory productivity (Hopp and Spearman 1996).

However, despite these flourishing developments, there remains a lack of comprehensive understanding of manufacturing systems. For example, JIT claims inventory to be the "root of all evil." Thus, many firms pursued WIP (work-in-progress) reduction programs in the 1980s. Some firms, which had high WIP level at the beginning, easily benefited from this attempt. Therefore, when discussing JIT, there was a tendency to emphasize its benefit for reducing the overall WIP, reducing costs and increasing flexibility, but to ignore the negative effects, such as lower processing system utilization. It is difficult to know how much WIP should be removed to reduce the cost, while maintaining enough WIP to prevent systems from idling due to interruptions of the upstream machines. Fully understanding this trade-off requires a more fundamental understanding of variability in manufacturing practice.

A fundamental issue for any factory is its cycle time at a given throughput rate. Even today, a fundamental problem is to accurately determine the cycle time of a production line which is composed of tandem workstations, when the machines are subject to breakdowns, setups and preventive maintenance. It seems the only reliable

method for making this assessment is simulation. As we will see later, even attempting to determine the cycle time of a single machine system in practice may not be an easy task, because in practice machines may have complicated configurations and may be subject to various types of interruptions.

In modeling manufacturing systems, it is well known that exact analysis of the cycle time of a simple tandem queue, which is only composed of two single servers, is difficult in general. Therefore, approximations are commonly used to estimate the system cycle time. Obviously, the performance analysis of queueing networks will be even harder.

It would be nice to be able to measure the performance of manufacturing systems exactly. However, a full description of a practical system is composed of a tremendous number of details. Some of them will have great impact on the system performance, but many of them may not. If we try to have an exact model considering all the details, we may lose sight of the fundamental behavior. Since determining the performance exactly is difficult, approximate methods are generally adopted by making assumptions to ignore or simplify some details. Which details should be ignored and which ones should be considered? The choice will directly impact the quality of approximation results.

1.1 The Underlying Structure

In practice, although two distinct systems are seldom exactly the same, they may share some common properties or structures. Those common properties can be classified as physical, behavioral or characteristic. We call a common property an underlying structure if the property is not directly observed or easily obtained.

Physical properties are the most easily observable. They refer to the common physical structures possessed by systems of the same kind. Examples are machines and buffers in manufacturing systems or pick-up/delivery locations in distribution systems.

Behavioral properties are the measured performance of the system. Examples are the inter-arrival times, queueing times and inter-departure times of jobs, utilizations of machines or traveling distances and times of vehicles. Because behavioral properties represent system performance, they can be used as the parameters in a mathematical model. Although the behavior properties are observable, sometimes, due to the complicated situations in practice, a complete and systematic description of the behavioral properties may not be easily obtained.

While the previous two are observable, the characteristic properties are inferred, and commonly used to describe the relations among behavior properties. Equations or constraints of a mathematical model belong to this category, but it can be more general, such as a theorem or property governing those equations. For example, the random walk is a basic model underlying many other models. It is embedded in the standard models of queueing theory, storage theory and time series analysis (Resnick, 2005). Another example is Little's law (Little 1961), which is fundamental in general queueing systems and describes the relationship among cycle time, throughput rate and WIP in the steady state.

Characteristic properties are often inferred with additional assumptions. For example, the independence assumption associated with the decomposition approach is commonly seen when dealing with large complex systems. As another example, exponential service times and inter-arrival times are assumed in Jackson networks, and lead to an exact product-form solution. When these assumptions hold, the system behaves as if each server acts independently in the steady state.

In addition to the independence assumption, other assumptions also can be seen. For example, when we assume the arrival process is Poisson, Wolff's PASTA (1982) tells us that the arrival jobs see time averages. The M/G/1 model proposed by Pollaczek (1932) and Khintchine (1932) determines the average cycle time in a single server exactly. If the servers are assumed to be in heavy traffic and the central limit theorem can

be applied properly, Brownian motion can be used to approximate the behavior of a single queue or a general queueing network.

In a single server queueing system, the server and buffer are the physical properties; the queueing time, capacity and throughput rate are the behavioral properties; Little's law, which describes the relation among WIP, cycle time and throughput rate is an example of the characteristic property. If we further assume the server is operated in heavy traffic and the central limit theorem can be applied properly, queueing time can be estimated by Kingman's G/G/1 approximation (Heyman 1975). Therefore, Kingman's approximation is an example of the characteristic property with additional assumptions.

Additional assumptions are commonly seen in searching for the characteristic properties. A fundamental question to ask is if they are appropriate. We say an underlying structure is powerful if it can generate needed results without making strong assumptions, i.e., assumptions that are unlikely to be satisfied in practice. Therefore, Little's law is a powerful underlying structure, since it can be applied to many general queueing systems without strong assumptions. Due to the Palm-Khintchine Theorem, the M/G/1 model can be also viewed as a powerful underlying structure. A powerful underlying structure can be used to simplify the unimportant details and let us focus on the major ones, which have greater impacts on system performance.

On the other hand, the strong assumption of exponential service times limits the applicability of Jackson networks in practical systems. Inspired by Jackson networks, the parametric-decomposition approach assumes each workstation in a queuing network is stochastically independent. This approach assumes the departure process is renewal and can be captured by the first two moments of its distribution. It focuses on how to analyze the behavior of each workstation in detail (considering batching, dispatching and interruptions, etc.) and ignores the correlations among workstations. However, as we will see, stochastic independence is a strong assumption especially when service time variability is small. Due to the different machine cost and customer expectation for

shorter cycle time, not all machines are designed to be operated in heavy traffic. Therefore, Brownian motion may not be a good assumption in practical manufacturing systems.

In the approaches that require strong assumptions, although we may conduct exact analysis and obtain precise results, the results may not be accurate as we will see in Chapter 5 and 6. We call those approaches with strong assumptions indirect approaches, since they model the behavior of practical systems by making strong assumptions at the beginning.

When using indirect approaches, we should be cautious about our original goal. If the original goal is to model the performance of an entire manufacturing system, making strong assumptions and solving the resulting model exactly may not bring us to the original goal which we want to attain. Suri, Sanders & Kamath (1993) comment, "Ignoring the ultimate needs of the manufacturing enterprise can lead to a whole generation of models and journal publications that become irrelevant to the very subject they purport to assist."

Determining system performance exactly is hard. Thus, our goal becomes to find an approximate model which is sufficiently accurate. Identifying the powerful underlying structures of manufacturing systems plays a key role in this aspect. The challenge is how to identify the powerful underlying structures. In the following chapters, we will explore the powerful underlying structures at different levels of manufacturing systems, from a single machine to a manufacturing system. Each chapter can be viewed as a specific example, which illustrates how to explore the underlying structures, and the power of identifying those useful underlying structures in manufacturing systems.

1.2 Dissertation Outline and Contributions

Manufacturing systems have been studied for a long time. Because randomness (from interruptions, service times and inter-arrival times) is embedded in almost every manufacturing system, any model which claims to credibly describe the behavior of manufacturing systems should have the ability to take stochastic effects into account.

Queueing theory, established by A.K. Erlang (1909), is a well-known method for evaluating the performance of manufacturing systems. Many important contributions have been made in applying queueing theory to manufacturing systems. Suri et al. (1993) gave a comprehensive review of applications of queueing theory in the performance evaluation of production networks. Buzacott and Shanthikumar (1993) discussed the applications of queueing theory in manufacturing systems and developed queueing models for various manufacturing environments. Papadopoulos, Heavey and Browne (1993) applied queueing network modeling, simulation modeling and optimization models to address the design and operational problems in manufacturing systems. Hopp and Spearman (1996) gave good explanation for variability trade-offs based on Kingman's approximations. They also introduced approximate models for many practical situations in manufacturing systems, such as transfer batches, process batches, and CONWIP systems.

The objective of this dissertation is to gain deeper understanding of manufacturing systems by identifying and exploring their underlying structures, mainly with the help of queueing theory. The first part of this dissertation focuses on the performance of a single machine system, and the second expands the scope to a manufacturing system.

The manufacturing context for this work will be semiconductor wafer fabrication facilities, or "fabs". The semiconductor industry has developed a standard for describing process tool states – SEMI E10 (2001) – which will form the basis for characterizing events for queueing analysis.

Queueing theory is a powerful tool to understand the performance of manufacturing systems. However, when we apply queueing models to a real production system, even for a single machine, many practical issues are encountered (see, e.g. Wu et al., 2007). A real machine is subject to various kinds of interruptions, such as breakdowns, setups and routine maintenance, etc. In practice, those interruptions can be time-based or run-based, and preemptive or non-preemptive. One of the main issues in analyzing performance is to choose the right queueing model when machines are subject to various kinds of interruptions.

Furthermore, real machines possess various configurations. In fabs, for example, wet benches are composed of a series of tanks, furnaces can be loaded four lots in a batch, and photolithography machines are combinations of tracks and scanners. The practical issues include clarifying the relations between queueing terminology and industry data standards, and approximating performance for various tool configurations.

In part I, we start with the issues in single machine manufacturing systems. In practice, machines are subject to several kinds of interruptions, and the queueing models should be appropriate for the specific properties of its interruptions. The pioneers in queueing theory have already derived many well known theoretical models. The crucial question here is about when to use each particular model, considering the complicated situations that arise in practice. Correct and comprehensive classification of interruptions is the first step towards the goal. Gaver (1962), Hopp and Spearman (1996) and Adan and Resing (2001) have proposed classifications of interruptions on the shop floor. By extending and unifying their results, a comprehensive classification will be proposed which considers all the events listed in SEMI E10 (2001).

The contribution of Chapter 2 is not only the development of integrated models, but also the classifications of currently known queueing models based on the structure of interruptions. In the classification, we group similar events in a way that supports Poisson driven modeling. In particular, we propose to model preventive maintenance as well as

other time-based non-preemptive interruptions by M/G/1 priority queues (Cobham 1954). In addition to the specific queueing model for each type of interruptions, an integrated model, which considers the occurrence of all interruptions, also is given in Chapter 2.

A powerful underlying structure presented in Chapter 2 is the decomposition property of time-based events, which allows us to first focus on run-based events only, since the difference between time and run-based events can be treated separately. This decomposition property is very important, since it allows the application of Kingman's G/G/1 approximation, which is appropriate for run-based interruptions, to situations with time-based interruptions.

In Chapter 2, the role of effective process time (EPT), introduced by Hopp and Spearman (1996), is compared with the concept of a generalized service time, or GST. We clarify potential confusion caused by differences between the concepts, and give necessary conditions for the proper use of EPT or GST. Furthermore, the two different interruption models, downtime events (ample resource cases) and resource contention problems, have been compared. The models for resource contention problems are much more complicated but could be approximated by the models of downtime events.

Batching, another important practical issue in manufacturing systems will be addressed in Chapter 3. Batching plays an important role in most factories, since it can lead to inefficiency if not treated with special care. There are two types of batching on the shop floor, process batches and transfer batches, where process batches can be further classified as serial batches and parallel batches. The G/G/1 based approximate models for parallel batch processing have been proposed by Bitran and Tirupati (1989a). By carefully examining the underlying structure of existing approximate models, an improved model for parallel batch systems is proposed. The computation of the new model is still simple and fast, but it gives better approximation by reducing the systematic error in earlier models caused by the dependence between queueing time and wait-to-batch time.

In Part II, the behavior of a factory will be addressed. We begin by discussing motivations and assumptions and, then turn our attention to the underlying structures of queues in series. Based on previous research (Jackson in 1957 and Friedman in 1965, etc.), approximate models are developed based on the insight from two special models: ASIA (all see initial arrivals) systems, and fully coupled systems.

The behavior of a simple tandem queue, which is composed of only two single servers, is addressed in Chapter 5. Since the simple tandem queue is the basic element of production lines, we want to understand the characteristics of production lines through first understanding the properties of simple tandem queues. In Chapters 5 and 6, the new models to approximate the tandem queue queueing times are proposed by taking advantage of the powerful underlying structure, the intrinsic ratio, of tandem queues. Based on the simulation results, the queueing time estimation errors caused by the stochastic independence and Brownian motion assumptions can be large, even in heavy traffic. The proposed models, which exploit the important underlying structure of tandem queue show that this error can be effectively reduced.

The dependence caused by the non-renewal departure process in manufacturing systems is intricate. It is the main reason which prevents us from analyzing general tandem queues exactly. However, based on the identified underlying structure, we propose a way to analyze the dependence among tandem queues from the view point of ASIA systems in Chapter 6.

Finally, in Chapter 7, the performance of a manufacturing system will be addressed. Rather than going into detailed analysis of each workstation, we take a macroscopic approach to gauge the variability of manufacturing systems by taking advantage of the powerful underlying structures identified in the previous chapters. This new model can describe the performance curve much more precisely than the previous models. Since variability can be used to describe the trade-offs between cycle time and

utilization, by quantifying the variability of manufacturing systems via historical cycle times, we can compare factory performances through their corresponding variabilities.

PART I

BEHAVIOR OF SINGLE MACHINE SYSTEMS

CHAPTER 2

BEHAVIOR OF A SINGLE MACHINE SYSTEM SUBJECT TO INTERRUPTIONS

Real world problems come first. Mathematical modeling comes next. Theory and algorithms follow as needed. ~ Dantzig

2.1 Introduction

One hundred years ago, the first paper on queueing theory was published by a Danish engineer, A.K. Erlang (1909), who worked in a telephone company. The early development of queueing theory was highly related to applications in teletraffic networks and mostly covered systems with exponential and deterministic service time distributions. In the mid-twentieth century, researchers began to consider more elaborate models, such as systems with general service times as well as systems with service interruptions. Those developments provided a solid foundation for the application of queueing theory to manufacturing systems.

In applying queueing theory to a single machine manufacturing system, both M/G/1 and G/G/1 models play important roles. For the M/G/1 queue, the Pollaczek-Khintchine (P-K) formula determines average cycle time in a single server with arrivals distributed according to a Poisson distribution. The formula was developed by both an Austrian-French mathematician, Felix Pollaczek (1932), and a Russian mathematician, Aleksandr Khintchin (1932). The M/G/1 queue is important in practical manufacturing systems, since the service time is not exponentially distributed in general, but the arrival process could be approximated by a Poisson process when each workstation is fed by multiple independent upstream workstations.

The exact M/G/1 models have direct influence on the derivation of G/G/1 approximations. The formal study of the G/G/1 model (in the notation of Kendall (1953)) can be traced back to Lindley (1952) in the early 1950s and a complete solution of the equilibrium queueing time distribution was obtained by Smith (1953). However, the solution is not easy to evaluate in general. Kingman (1965) studied the heavy traffic approximation for G/G/1 systems. By observing the upper bound derived by Kingman (1962a), Heyman (1975) derived the heavy traffic approximation for G/G/1 queue by diffusion models. Kramer and Lagenbach-Belz (1976) proposed an improved G/G/1 approximate model based on a purely heuristic extension of Pollazcek-Khintchine formula and Heyman's results. Kingman's pioneering work in the G/G/1 approximation plays an important role in applying queueing theory to manufacturing systems.

Interruptions are common in practical manufacturing systems. Queueing models which incorporate the influence of interruptions have been studied since the 1950s (e.g., White and Christie (1958)). Gaver (1962), Keilson (1962) and Avi-Itzhak et al. (1963) gave preliminary studies on the M/G/1 models for queues subject to service interruptions. Hopp and Spearman (1996) classify interruptions as preemptive and non-preemptive. Adan and Resing (2001) gave M/G/1 models for both unreliable machines and the machines with setup times.

Because of the role of manufacturing in maintaining a firm's competitiveness (Hayes et al. 1988), effective analysis and improvement of manufacturing systems have become important activities in the survival of modern firms. Queueing theory plays an important role in this regard. In the early 1990s, Suri et al. (1993) gave a comprehensive review of applications of queueing theory to the performance evaluation of production networks. Buzacott and Shanthikumar (1993) discussed the applications of queueing theory in manufacturing systems and developed queueing models for various manufacturing environments, such as assembly lines, Kanban systems and flexible machine systems. Hopp and Spearman (1996) gave good explanation for variability

trade-offs based on Kingman's approximations. They also introduced approximate models for many practical situations in manufacturing systems, such as transfer batches, process batches, and CONWIP systems.

The main purpose for developing those queueing models was to solve real problems in manufacturing systems. However, as pointed out by Wu et al. (2007), when we apply queueing models to a real production system, even for a single machine, some practical issues are encountered. A real machine is subject to various kinds of interruptions, such as breakdowns, setups and routine maintenance, etc. In practice, those interruptions can be time-based or run-based, and preemptive or non-preemptive. Thus, a critical issue in performance evaluation is to choose the right theoretical queueing model when machines are subject to various kinds of interruptions.

Hopp and Spearman (1996) explain how to apply G/G/m approximations to evaluate the performance of manufacturing systems by defining service time (ST) using the notion of effective process time (EPT), which accounts for process time, setup, breakdown, and all other operational delays due to variability effects. However, Wu and Hui (2008) pointed out that, when there are time-related disruption events, the effective process time defined for an M/M/1 queueing model cannot be measured precisely in practice but can only be estimated statistically.

Motivated by this issue, Wu et al. (2007) classified all the activities defined in SEMI E10 into Type-I events, which are WIP (Work-In-Progress) related events and Type-II events, which are time-related events. However, they did not differentiate between interruptions and natural variability, and combined both into Type-I events. In this chapter, with a better understanding of system behavior, we provide a more complete classification of shop floor activities. Considering the naming rule from FabSim (Billings, 2006), we call the WIP-related events "run-based events" and denote time-related events by "time-based events". This classification also has been proposed by Buzacott and

Hanifin (1978) for the analysis of automatic transfer lines, although they do not further classify preemptive and non-preemptive events as the cause of interruptions.

Based on our proposed classification for the factors which affect queueing times, the corresponding queueing model for each type of event is introduced. This new classification has been compared with the classification described in SEMI E10 (2001), which classifies events on the shop floor from the view point of machine efficiency. An integrated queueing model, which considers all types of interruptions, has been derived based on the classification.

Knowing how to derive the model is important, but knowing how to use the model to solve the problems which originated it is also essential. In this chapter, we focus not only on the derivation of models, but also on how to use the models to solve a practical problem by observing the relationship among the models and the insight behind the models. For example, by identifying the constant gap between time-based and run-based interruptions, we demonstrate how to apply Kingman's approximation to a time-based interruption, rather than only to the run-based interruption.

The content of this chapter is written for both practitioners and researchers. For practitioners, we identify the queueing models appropriate for different types of interruptions. Through this, we hope practitioners can treat this chapter as a dictionary and use the correct queueing model in each identified situation. For researchers, we give derivations of each model, explain the gap between time-based and run-based events, and give derivations of the integrated models.

This chapter is organized as follows. We classify different events from the perspective of queueing theory in Section 2.2 and clarify the definitions of some important terms and concepts in 2.3. We propose the queueing models suitable for each specific category of interruption in Section 2.4. In Section 2.5, the relationships among queueing models are modeled and an integrated model is proposed. In Section 2.6, the model for resource contention problems is discussed. Simulation experiments are

reported in Section 2.7. In Section 2.8, the comparison between two classifications, SEMI E10 and queueing theory, is presented. Concluding remarks, including directions for future research are given in Section 2.9.

2.2 The Structure of Interruptions

Queueing theory predicts system delay performance under the influence of randomness. The randomness comes from either natural variability of inter-arrival and service times, or from interruptions. Natural variability refers to the randomness caused by the system performing its *intended functions*. These sources of randomness come from natural properties designed into, or inherent to the system, such as job release patterns, service time differences caused by product mix, or transfer time differences caused by robot motions, etc. Therefore, natural variability will not decrease the availability of machines. Interruptions are events whose occurrence prevents the system from performing its *intended function*. The negative impact of interruption can be longer cycle time, less capacity, or both. Examples of natural variability and interruptions are summarized in Figure 2.1.

When there are no interruptions and both service and inter-arrival times are exponentially distributed, the M/M/1 model is appropriate. If times are generally distributed, the G/G/1 model can be used. Later, we will show that, even in the presence of run-based interruptions, the above models are still valid in a certain sense. However, when time-based interruptions are present, the above models are not exact anymore and their estimates of performance could be quite misleading.

Interruptions are inherent in any manufacturing system and can be either preemptive or non-preemptive. They are caused by the interactions between a process and an event, and have negative impacts on productivity. Based on SEMI E10 (2001), downtime is “the time when the equipment is not in a condition, or is not available, to

perform its intended function." Examples are breakdowns, experiments, preventive maintenance (PM) and setups. There are two types of down states: unscheduled down time and scheduled downtime. Since scheduled downtime implies an ability to control when downtime happens, it is usually viewed as non-preemptive. Since unscheduled downtime is not controllable, it is usually preemptive. Furthermore, failure is defined as "any *unscheduled* downtime event that changes the equipment to a condition where it cannot perform its intended function." Therefore, in this chapter, we specifically define failures as preemptive events. By definition, failures will decrease the availability of machines, but scheduled downtime events will not.

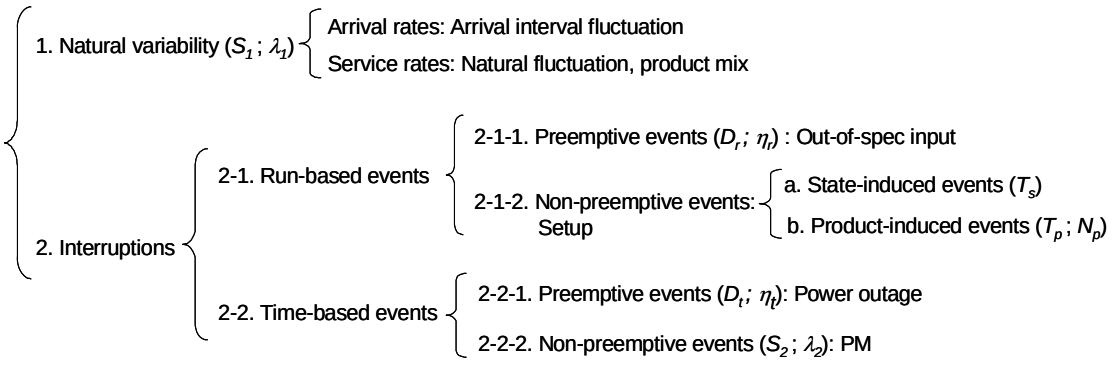

Figure 2.1 Classification of factors which affect queueing times

Another approach, proposed by Wu and Hui (2008), is to classify interruptions as run-based or time-based events. Run-based events are associated with WIP, while time-based events can occur anytime, whether or not WIP is present. For example, breakdowns caused by power outages are time dependent, and should be classified as time-based events, while setups due to differences in recipes constitute run-based events. Other typical examples are:

- Run-based events: out-of-spec input, setup
- Time-based events: power outage, PM, experiment

It should be noted that the service time variation induced by product mix is not a run-based event, since it is not an interruption, and should be considered part of natural variability. A notable difference between natural variability and interruptions is the nature of occurrence and impact. For natural variability, the influence on jobs only occurs in processing. Therefore, it must be run-based. An interruption will impact jobs only when the event occurs. Therefore, the impact of interruptions can be either time-based or run-based, depending on the nature of occurrence. Because the influence on jobs from natural variability only occurs in processing, we can model their impact by adjusting service time or inter-arrival time distributions to account for them. Furthermore, run-based events can be considered a special case of time-based events, since time-based events can occur at any time, but run-based events can only occur during processing.

Both run-based and time-based events can be further classified as preemptive or non-preemptive. A preemptive event can occur anytime during processing, but a non-preemptive event can only occur before or after processing. Therefore, a run-based non-preemptive event can only before job processing starts or after job processing ends, since it cannot preempt processing yet is associated with the processing of jobs. Setups are typical examples of this type. In addition to run-based non-preemptive events, out-of-spec inputs are run-based preemptive events, power outages are time-based preemptive events, and PM is a time-based non-preemptive event. The different classifications are summarized in Figure 2.1.

Run-based non-preemptive events are further classified as state-induced and product-induced events. State-induced events correspond to machine changing state either from busy to idle or idle to busy. As an example, consider a machine that goes into a “sleep mode” when it is idle, and requires some warm up time when it returns to production mode. It experiences a run-based non-preemptive state-induced interruption.

Product-induced events correspond to switching machine settings for different products. For example, a machine may need some setup time when switching from one

recipe to another. There is a fundamental difference between these two types of events: state-induced setups will vanish when a machine is fully utilized, since no state change occurs if a machine is always busy. However, product-induced setups are determined by external customer demands. Thus, they cannot be completely avoided. We may alter the frequencies of product-induced setups by changing scheduling rules, but we simply cannot run one product all the time if customers demand more than one product.

Capacity for a single machine system is the maximum throughput rate of the machine. The previously specified difference between product-induced and state-induced events leads to different impacts on capacity. Product-induced events have impacts on both cycle time and capacity, but state-induced events have impacts only on cycle time and not capacity, as will be explained in Section 2.3.

In Figure 2.1, all interruptions are either (a) downtime events, or (b) resource contention problems. For a downtime event, we assume that the resources needed to recover from the interruptions are always available (ample resource case). However, for a resource contention problem, the resources are limited and are sometimes unavailable. Basically, resource contention is caused by activities of other entities or machines. The machine is forced to be idle (but still in production mode) when the required resource is occupied by some other machine. Examples of such resources can be operators, engineers, mask sets, support tools or parts. If those resources are shared among multiple machines and we do not have full control of the occurrences of interruptions, resource contention will occur. Therefore, operator availability does not belong to any type of downtime events. It affects system performance through downtime events, such as setups or load activities, but must be modeled by resource contention models instead of downtime event models.

2.2.1 Comparison with Previous Work

Although the classification proposed in Figure 2.1 is inspired by the direct observation from SEMI E10, the idea to classify different types of interruptions in manufacturing systems is not new. Several researchers have classified interruptions from the view point of queueing theory.

Gaver (1962) first distinguished interruptions as active and independent, where each case is further classified as preemptive or postponable. This approach is very similar to our classification. However, while this approach is theoretically sound, it does not take into account the practical situations in real manufacturing systems. Gaver did not further classify the active-postponable cases as state-induced or product-induced, but simply modeled them as Poisson arrivals during processing times. Furthermore, for the independent-postponable cases, he assumes no interruptions can occur during an interruption, and all interruptions have high-priority, which is not always true in practice.

Avi-Itzhak and Naor (1963) presented the extensions of M/G/1 queueing models for five different types of interruptions. Compared with the classifications introduced in this paper, their Model A is indeed the same as time-based preemptive events, and Model B is the same as run-based preemptive events. However, except for the above two models, the rest of their models are more likely to be applied to telecommunication instead of manufacturing systems.

Hopp and Spearman (1996) classify interruptions as preemptive and non-preemptive, but fail to differentiate time-based and run-based interruptions. Adan and Resing (2001) gave M/G/1 models for two special cases of interruptions: unreliable machines and machines with setup times. However, their unreliable machine models only consider the time-based preemptive events, and their setup models only consider the run-based non-preemptive state-induced setups.

All the above classifications have been summarized in Table 2.1. Although queueing models for various interruptions have been addressed before based on each

classification, a systematic way to compare and incorporate all models in an integrated structure of interruptions in practical manufacturing systems has not been previously described.

Table 2.1 Comparison of different classifications

Classification from Fig. 1	Gaver	Avi-Itzhak and Naor	Hopp and Spearman	Adan and Resing	Wu, et al.
2-1-1	●	●	●		●
2-1-2.a				●	●
2-1-2.b			●		●
2-2-1	●	●		●	●
2-2-2	⊘				●

2.3 Definitions

The three fundamental parameters of queueing models are mean service time ($1/\mu$), mean inter-arrival time ($1/\lambda$) and number of servers. While inter-arrival time and server count are clearly defined in manufacturing systems, the definition and measurement of service time (ST) sometimes is not completely clear in the presence of interruptions. The role of service times in queueing theory is closely related to the concept of capacity, the maximum throughput rate of a system. In the simplest queue, when there are no interruptions, the reciprocal of service time (i.e. service rate) is the capacity of a workstation. However, when there are interruptions, service times may be more difficult to define.

When there are no interruptions, mean and variance of service time are only affected by the nature of service time itself. In order to define service time when interruptions occur, we must first consider the meaning of system capacity. Capacity is the maximum throughput rate of a system, and it can be achieved when WIP is always available. In this situation (when a machine is always busy), a preemptive event can only

occur during processing. Therefore, in particular, time-based preemptive events can only occur during machine processing. In other words, in determining capacity, we do not need to consider the impact from those time-based preemptive events which occur when the machine is not busy. Furthermore, no state-induced events occur when a machine is always busy. Thus, in assessing capacity, there are four types of events that must be considered: run-based preemptive events (2-1-1), time-based preemptive events occurred during processing (2-2-1), run-based non-preemptive product-induced events (2-1-2.b) and time-based non-preemptive events (2-2-2). To simplify the notation, we merge the first two types of events (2-1-1 and 2-2-1), and call them preemptive events (which occur during processing). Now, there are only three groups of events that must be considered: preemptive events (2-1-1 and 2-2-1), run-based non-preemptive product-induced events (2-1-2.b) and time-based non-preemptive events (2-2-2). The first one can be either run-based or time-based, the second one is run-based and the third one is time-based.

Considering the situation when there is infinite WIP in queue; the machine state at all times will be either:

1. Processing jobs, handling preemptive events (2-1-1 and 2-2-1) or handling run-based non-preemptive product-induced events (2-1-2.b), or
2. Handling time-based non-preemptive events (2-2-2).

In each state, machine capacity is being consumed. The first type of capacity consumption is caused by processing jobs, while the second type is caused by time-based non-preemptive events. In order to make the system stable, the total workload from both types must be less than the total capacity.

The durations of the two states are generally assumed to be independent. Furthermore, for the second state, we assume the duration is determined by the non-preemptive event itself and is independent of the length of service time. However, the

duration of the first state is determined by the run-based non-preemptive product-induced events (if any) and service time, as well as the preemptive events occurring during the service time. With separate consideration of the second state (which is purely caused by interruptions, and will be discussed in Section 2.4.4), machine capacity will be determined solely by the duration of the first state. Since service time in queueing models is the reciprocal of capacity, we need to extend the notion of service time based on the duration of the first state.

An important generalization of service time is the generalized service time (GST), which reflects the capacity of a workstation under the influence of interruptions. Based on the above discussion, GST is defined as

$$GST = \textit{Job departure time} - \textit{The time epoch when the job first claims capacity of the machine,} \quad (2.1)$$

where job departure time is the time that a job releases the machine capacity. A job claims capacity of a machine if

1. the job is present at the machine,
2. the preceding job has released the machine capacity,
3. the machine is ready to process this job, or is ready to be engaged in a product-induced event.

Based on the previous analysis, capacity is consumed only when the machine is in one of the two states. Since the second state is caused purely by interruptions, only the first state, which is triggered by processing jobs, should be considered in defining service time. Based on the conditions of the first state, a job can claim machine capacity either when the machine is ready to process this job or is ready to be engaged in a product-induced event (2-1-2.b).

Therefore, if a job arrives when the machine is down, it cannot claim capacity until the machine is ready for production, or to be engaged in a product-induced event. It should be noted that the setup times caused by product-induced events are counted into GST, but the setup times caused by state-induced events are not.

Based on the above definition, GST is the summation of product-induced setup time (if any), service time, and the downtimes of all preemptive events occurring during that service time,

$$G = S + \sum_{i=1}^{N(S)} D_i + T_p, \tag{2.2}$$

where S is service time, N(S) is the number of preemptive events (e.g. breakdowns) during S, D_i is the i-th downtime, and T_p is the duration of a run-based non-preemptive product-induced event experienced by a job. Here we assume there is no preemptive interruption during setups, or, if there is, those interruptions can be modeled by the distribution of T_p.

The definition of generalized service time plays an important role in deriving the integrated models in Section 2.5. Based on the classifications in Table 2.1, generalized service time sometimes has been presented in a simpler form. Without considering the impact from run-based non-preemptive events, both Gaver (1962) and Adan and Resing (2001) define GST as

$$G = S + \sum_{i=1}^{N(S)} D_i, \tag{2.2a}$$

which they called generalized processing time, and used it to derive the queueing models for time-based preemptive events. Although Eq. (2.2) and (2.2a) are similar, Eq. (2.2) considers all the interruptions in Figure 2.1, and recognizes the impacts on capacity from both state-induced events (2-1-2.a) and product-induced events (2-1-2.b).

Capacity (or maximum throughput rate) is well defined when the machine is always busy. However, the generalized service time defined in Eq. (2.1) and (2.2) is not

limited to the situations where the machine is always busy. For example, the service time of an M/M/1 system is 1/μ, where μ is the maximum throughput rate or capacity. Although μ is the throughput rate when the machine is always busy, service time is the same (1/μ) for any utilization level.

When both service time and downtime are generally distributed, and both the job arrival intervals and mean time between preemptive events are exponentially distributed, the mean of GST is

$$\begin{aligned} E(G) &= E(S + \sum_{i=1}^{N(S)} D_i) + E(T_p) \\ &= \int_0^\infty \sum_{n=0}^{\infty} (x + x\eta E(D)) e^{-\eta x} \frac{(\eta x)^n}{n!} f_s(x) dx + E(T_p) \\ &= E(S) + E(S)\eta E(D) + E(T_p), \end{aligned} \tag{2.3}$$

where $1/\eta$ is the mean time between preemptive events, and D is downtime for the preemptive events which occur during the service period.

Availability (A) is defined as

$$A = \frac{m_f}{m_f + m_r}, \tag{2.4}$$

where m_f is the mean time between failures (MTBF) or the mean time between preemptive events, and m_r is the mean downtime or mean time to repair (MTTR). Based on the same assumptions of Eq. (2.3) (i.e. both service time and downtime are generally distributed, and both the job arrival intervals and mean time between preemptive events are exponentially distributed), Eq. (2.4) becomes

$$A = \frac{m_f}{m_f + m_r} = \frac{1/\eta}{1/\eta + E(D)}. \tag{2.4a}$$

Therefore, based on Eq. (2.4a), Eq. (2.3) can be restated as

$$E(G) = E(S)/A + E(T_p). \tag{2.3a}$$

Furthermore, if both m_f and m_r are exponentially distributed, and μ is the mean service rate, availability and $E(G)$ can be expressed as

$$A = \frac{m_f}{m_f + m_r} = \frac{1/\eta}{1/\eta + 1/\theta} = \frac{\theta}{\theta + \eta}, \tag{2.4b}$$

$$E(G) = 1/(\mu A) + E(T_p), \tag{2.3b}$$

where $1/\eta$ is the mean time between preemptive events, and $1/\theta$ is the mean time to recover from those events.

Using the definition of GST, cycle time (CT) can be explicitly defined as

$$CT = QT + GST, \tag{2.5}$$

where QT is queueing time.

In the above analysis, the definition of service time is based on the concept of capacity. If we neglect this fundamental insight and instead assume capacity is derived from the concept of service time, this oversight will lead to imprecise estimations.

To illustrate this, consider the concept of effective process time (EPT), defined by Hopp and Spearman (1996): the total time "seen" by a job at a station. It does not matter whether the job is actually being processed or is being held up because the machine is being repaired, undergoing a setup, rework, or waiting for its operators.

The definition of EPT is basically the same as the definition of GST given in Eq. (2.1), except for the conditions to claim machine capacity. For EPT, we only need the first two,

1. the job is present at the machine,
2. the preceding job has released the machine capacity.

Since the third condition of GST is dropped, the machine may or may not be ready to process jobs when the capacity is claimed. Thus, EPT is the same as GST only when:

(a) all interruptions are run-based,

(b) all run-based non-preemptive events are product-induced.

If condition (b) does not hold, there are state-induced run-based non-preemptive events. EPT will incorporate the event time into its duration, which will lead to the miscalculation of capacity, since state-induced events should not decrease capacity. Furthermore, with higher system utilization, there are fewer occurrences of state-induced events, and EPT becomes dependent on system utilization. Independence of service times and arrival intervals is a key assumption in queueing theory approaches to modeling manufacturing systems. If it does not hold, the model will become complicated or even intractable. Therefore, this is a critical theoretical issue for the EPT approach.

For condition (a), if there are time-based interruptions, an arriving job may be blocked by an interruption. This waiting period will be counted into EPT (but not in GST). Since the interruptions have higher probability to block a job when the job arrival rate is high, the value of EPT depends on the machine utilization. This dependence contradicts the assumption that service time is independent of utilization.

Therefore, in order for queueing theory models based on EPT to be valid, the preemptive events in EPT can be run-based only. If all preemptive events are run-based, and all run-based non-preemptive events are product-induced, a job can always claim capacity of the machine instantly if it arrives when the machine is not busy. However, this is not the case in general, since the preemptive events can be time-based. Even if a job arrives when the machine is not busy, it may wait if the machine is down when it arrives.

When conditions (a) and (b) described above hold, EPT is the same as GST. Thus, from (2.3a), we have:

$$E(EPT) = E(S)/A + E(T_p). \quad (2.3c)$$

Hopp and Spearman (1996) develop queueing models based on Eq. (2.3c). As argued here, those models are exact only when situations (a) and (b) hold. This could lead to errors if time-based events are common. For example, if a job arrives when there is no job in queue and the machine is down (due to a time-based event), based on condition (1) and (2), it should claim machine capacity immediately. However, as we explained, this will cause the dependence between EPT and utilization, and contradicts the conditions of claiming capacity of GST in Eq. (2.1)

When there are time-based events, the concept of EPT should be used carefully because, in general,

$$CT \neq QT + EPT.$$

If there is no time-based non-preemptive interruption, utilization (ρ) is defined as

$$Utilization = \lambda / Service\ rate,$$

where λ is the arrival rate and service rate is the maximum throughput rate (i.e. capacity) of the system. Therefore, when both service time and downtime are generally distributed, and both the job arrival intervals and mean time between preemptive events are exponentially distributed, based on Eq. (2.3), utilization of a single machine system can be expressed as

$$Utilization = \lambda E(G).$$

If there are no product-induced setup events, based on Eq. (2.3a),

$$Utilization = \lambda E(S)/A.$$

If there is no failure (i.e. no preemptive events), the above equation becomes

$$Utilization = \lambda E(S) = \lambda / \mu.$$

2.4 Queueing Models for Each Category of Interruption

Based on the classifications introduced in Section 2.2 and the definitions given in Section 2.3, we are now ready to derive queueing models for each category of downtime events. The run-based interruption models will be introduced in Section 2.4.1 and 2.4.2, and the time-based interruption models will be introduced in Section 2.4.3 and 2.4.4. Integrated models which consider all interruptions will be given in 2.5. Models for resource contention problems will be discussed in Section 2.6.

2.4.1 Models for Run-based Preemptive Interruptions

For run-based interruptions, we analyze the preemptive events (2-1-1) first and start from the simplest case, i.e., the M/M/1_Run-based preemptive event model. In this model, the machine can break down only when it is processing jobs. Jobs arrive according to a Poisson process with rate λ, and service times are exponentially distributed with mean $1/\mu$. The up and down times are also exponentially distributed with means $1/\eta$ and $1/\theta$. For stability, we assume that

$$\rho = \lambda / (\mu A) < 1,$$

where A is defined in Eq. (2.4b). Based on the property of "Poisson arrivals see time averages" (PASTA) proposed by Wolff (1982), an arriving job finds on average $E(L)$ jobs in the system and encounters $\eta E(L)/\mu$ breakdowns. Furthermore, each job, on arrival, sees the machine is already down with probability ρ(1-A) (The downtime being considered only occurs when the machine is being used). Therefore,

$$\begin{aligned} E(QT) &= \frac{E(L)}{\mu} + \eta\left[\frac{E(L)}{\mu}\right]\frac{1}{\theta} + \rho(1-A)\frac{1}{\theta} \\ &= \frac{E(L^q)+\rho}{\mu} + \eta\left[\frac{E(L^q)+\rho}{\mu}\right]\frac{1}{\theta} + \rho(1-A)\frac{1}{\theta} \\ &= \frac{1}{\mu}E(L^q)(1+\frac{\eta}{\theta}) + \frac{\rho}{\mu}(1+\frac{\eta}{\theta}) + \rho(1-A)\frac{1}{\theta} \end{aligned}$$

$$= \frac{E(L^q)}{\mu A} + \frac{\rho}{\mu A} + \rho(1-A)\frac{1}{\theta}, \tag{2.6}$$

where L is the number of jobs in the system (both waiting and processing jobs), L^q is the number of jobs in queue and A is availability from Eq. (2.4b). By Little's law (Little 1961), QT can be further simplified,

$$E(L^q) = \lambda E(QT), \tag{2.7}$$

$$E(QT) = \frac{\rho}{1-\rho}\frac{1}{\mu A} + \frac{\rho}{1-\rho}\frac{1-A}{\theta}, \tag{2.8}$$

and

$$E(CT) = E(QT) + E(G) = \frac{1}{1-\rho}\frac{1}{\mu A} + \frac{\rho}{1-\rho}\frac{1-A}{\theta}. \tag{2.9}$$

A more general case is the M/G/1_Run-based preemptive event model. The assumptions are basically the same as above except the service time and down time are generally distributed. Because service time and down time are generally distributed, we do not have the memoryless property anymore. Therefore, it is important to know if the service could be resumed after the downtime. In this chapter, we assume all the service time devoted to a job before the machine is interrupted is not lost and service is resumed once the machine recovers from failure.

The first and second moment of the service time are denoted by $E(S)$ and $E(S^2)$. The up time between two breakdowns is exponentially distributed with mean $1/\eta$. The first and second moments of the down time are denoted by $E(D)$ and $E(D^2)$. For stability, we assume

$$\rho_G = \lambda E(G) < 1.$$

By PASTA, an arriving job finds on average $E(L^q)$ jobs in queue, and a working job with probability ρ_G. Therefore,

$$E(QT) = E(L^q)E(G) + \rho_G E(R_G), \tag{2.10}$$

where

$$E(G) = E(S) + E(S)\eta E(D),$$
$$E(G^2) = E(S^2)\left[1+\eta \mathrm{E}(D)\right]^2 + E(S)\eta E(D^2),$$
$$\rho_G = \lambda E(G),$$
$$E(R_G) = E(G^2)/2E(G).$$

By Little's law, QT can be further simplified as follows,

$$E(L^q) = \lambda E(QT),$$

$$E(QT) = \frac{\rho_G E(R_G)}{(1-\rho_G)} = \left(\frac{1+c_e^2}{2}\right)\left(\frac{\rho_G}{1-\rho_G}\right)E(EPT), \qquad (2.11)$$

where c_e^2 is the squared coefficient of variation (SCV) of EPT, and

$$E(CT) = E(QT) + E(G).$$

The first equality of Eq. (2.11) is first given by Gaver (1962) in a slightly different form, which took transfer batch sizes into account. The second equality of Eq. (2.11) holds because we are only dealing with run-based preemptive events. Since Eq. (2.8) is a special case of Eq. (2.11), it can be shown that Eq. (2.8) and (2.11) are both consistent with the results given by Hopp and Spearman (1996). Detailed derivations are given in Appendix A. Although only the mean is given by Eq. (2.11), the variance of QT has been given by both Gaver (1962) and Hopp and Spearman (1996).

The Palm-Khintchine Theorem says, "The superposition of a large number of independent and renewal processes is approximately a Poisson process." In manufacturing, especially in wafer fabs, the Palm-Khintchine Theorem could be used to justify the assumption of Poisson arrivals, especially when the machine is fed by multiple upstream workstations, and each workstation is composed of multiple machines. Because of the multiple upstream machines and complex routing, we could have a large number of independent arrival processes. However, we should also keep in mind that those arrival

processes are non-renewal in general, so the conditions of Palm-Khintchine Theorem are not faithfully followed. A more detailed discussion on this topic will be given in Part II.

When there are multiple sources of failures, and the arrival processes are independent and renewal, assuming that MTBF is exponentially distributed is reasonable. Therefore, M/G/1 models could suffice for some situations in practice. For the more general situation, the G/G/1_Run-based preemptive model, we need to resort to approximations. The appropriate formulations are given in the next section together with the non-preemptive cases.

2.4.2 Models for Run-based Non-Preemptive Interruptions

In this section, we demonstrate different formulations for the case of run-based non-preemptive events (2-1-2). We first assume the machine needs a setup whenever it changes state from idle to production. For example, the machine is turned off when it is idle, and turned on again when a new job arrives, but restarting takes some time. Another example is the use of monitor wafers. For maintaining consistent process quality, it is common in semiconductor fabs that, after a machine has been idle, a few special wafers are run before a series of production wafers. Those special wafers are then sent for inspections, such as particle counts or film thickness measurement. Those special wafers, called monitor wafers, will cause delay of production much like a job setup, or a run-based non-preemptive state-induced event.

For this type of event, we start from the simplest case, the M/M/1_Run-based non-preemptive queue with state-induced events. In this model, jobs arrive according to a Poisson process with rate λ, and service times are exponentially distributed with mean $1/\mu$. Setup times are exponentially distributed with mean $1/\theta$. For stability, we assume

$$\rho = \lambda / \mu < 1.$$

By PASTA, an arriving job finds on average $E(L^q)$ jobs in queue and sees a

working job with probability ρ. If the machine is busy when the job arrives, there is no setup, so the probability of setup is 1 - ρ. Therefore,

$$E(QT) = E(L^q)\frac{1}{\mu} + \rho\frac{1}{\mu} + (1-\rho)\frac{1}{\theta}. \tag{2.12}$$

Applying Little's law,

$$E(L^q) = \lambda E(QT),$$

and, with (2.12) we get

$$E(QT) = \frac{\rho}{1-\rho}\frac{1}{\mu} + \frac{1}{\theta}, \tag{2.13}$$

and thus:

$$E(CT) = E(QT) + E(S) = \frac{1}{1-\rho}\frac{1}{\mu} + \frac{1}{\theta}. \tag{2.14}$$

This expression also has been derived by Adan and Resing (2001) in a different way. From Eq. (2.13), the queueing time of an M/M/1_Run-based non-preemptive state-induced event model is just the queueing time of an M/M/1 model plus an extra setup time.

A more general case is the M/G/1_run-based non-preemptive queue with state-induced events. The assumptions are the same as above, except the service times and setup times are generally distributed. The first and second moment of the service time are denoted by $E(S)$ and $E(S^2)$. The first and second moment of the setup time are denoted by $E(T_s)$ and $E(T_s^2)$. For stability, we assume

$$\rho = \lambda E(S) < 1.$$

Assuming PASTA, an arriving job finds on average $E(L^q)$ jobs in queue and sees a working job with probability ρ. Otherwise, it arrives when the machine is either idle or in a setup phase. Therefore,

$$E(QT) = E(L^q)E(S) + \rho E(R_S)$$
$$+(1-\rho)\left[\frac{1/\lambda}{1/\lambda + E(T_s)} E(T_s) + \frac{E(T_s)}{1/\lambda + E(T_s)} E(R_T)\right], \qquad (2.15)$$

where

$$E(R_S) = E(S^2)/2E(S),$$

$$E(R_T) = E(T_s^2)/2E(T_s).$$

Combining Eq. (2.15) and Little's law, we get,

$$E(QT) = \frac{\rho E(R_S)}{(1-\rho)} + \frac{1/\lambda}{1/\lambda + E(T_s)} E(T_s) + \frac{E(T_s)}{1/\lambda + E(T_s)} E(R_T), \qquad (2.16)$$

and

$$E(CT) = E(QT) + E(S).$$

See also the notes by Adan and Resing (2001) for a different derivation, where the expected cycle time is derived directly. From Eq. (2.16), the queueing time of an M/G/1_Run-based non-preemptive state-induced event model is just the queueing time of an M/G/1 model plus an extra period caused by the setups. An important observation is that run-based non-preemptive state-induced events affect the original queueing formula by simply adding extra terms (in Eq. (2.13) and (2.16)).

$$Gap = \frac{1/\lambda}{1/\lambda + E(T_s)} E(T_s) + \frac{E(T_s)}{1/\lambda + E(T_s)} E(R_T), \qquad (2.16a)$$

Property 1 (*Decomposition property of state-induced events*).

For any run-based non-preemptive state-induced interruption, its queueing time is the same as the queueing time of the original system, which does not suffer from the state-induced interruption, plus an extra gap. This gap is upper bounded.

Recall that state-induced events only impact cycle time – not capacity. This can be verified by Eq. (2.13), (2.16) and the stability condition. Another interesting observation is that when λ approaches μ (i.e. ρ approaches 1), the queueing time of the original M/M/1 or M/G/1 queueing models increase without limit, but the extra setup times are bounded. Therefore, the cycle time at high utilization is dominated by queueing time instead of setup time.

In the second model (the run-based non-preemptive queue with product-induced events), we assume the machine needs a setup for product changeovers. The setups occur due to changes in the production process induced by switching products. This model is also introduced by Hopp and Spearman (1996). We describe this model here briefly for completeness.

For this model, the machine processes an average of N_p jobs between two consecutive setups, and the probability of doing a setup after any job is $1/N_p$. The product-induced setup times, P, have a mean of $E(P)$ (or t_p) and a standard deviation of σ_p. Since GST is the same as EPT in run-based non-preemptive product-induced event model, we have

$$GST = S + T_p = EPT,$$

$$E(G) = E(S + T_p) = E(S) + E(T_p)$$
$$= t_0 + t_p / N_p = t_e,$$

where

$$t_p = E(P),$$

T_p is the product-induced setup time experienced by a job, t_e is the mean of EPT and t_0 is the mean of service time. For stability, we assume

$$\rho = \lambda E(G) = \lambda (t_0 + t_p / N_p) < 1.$$

Comparing the stability conditions of the state-induced and product-induced models reveals that the setup times of product-induced models have direct impact on

capacity, while the setup times of state-induced models do not. This can explain why the durations of the state-induced events are not counted into the generalized service times as described in Eq. (2.2).

By PASTA, an arriving job finds on average $E(L^q)$ jobs in queue and sees a working job with probability ρ. It experiences a setup with probability $1/N_p$. Similar to Eq. (2.10), we have

$$E(QT) = E(L^q)E(G) + \rho_G E(R_G), \tag{2.17}$$

where

$$\rho_G = \lambda E(G),$$

$$E(R_G) = E(G^2)/2E(G),$$

$$E(G^2) = E(S^2) + 2E(S)E(T_p) + E(T_p^2).$$

Combining (2.17) and Little's law, we get,

$$E(QT) = \frac{\rho_G E(R_G)}{(1-\rho_G)} = \left(\frac{1+c_e^2}{2}\right)\left(\frac{\rho_G}{1-\rho_G}\right)E(EPT), \tag{2.18}$$

and,

$$E(CT) = E(QT) + E(G).$$

The second equality of Eq. (2.18) holds because we are only dealing with run-based non-preemptive product-induced events. Appendix A contains detailed derivations. The variance of QT has been given by Hopp and Spearman (1996).

2.4.2.1 Moving Away from the Poisson Arrivals

Since GST is the same as EPT in the above case, we can directly use the results from Factory Physics (Hopp and Spearman 1996). The mean and variance of EPT are as follows,

$$t_e = t_0 + t_p / N_p, \tag{2.19a}$$

$$\sigma_e^2 = \sigma_0^2 + \frac{\sigma_p^2}{N_p} + \frac{N_p - 1}{N_p^2} t_p^2. \tag{2.19b}$$

where σ_0 is the standard deviation of service time. By using the heavy traffic approximations (Heyman 1975), queueing time (QT) of G/G/1 queues can be estimated by Eq. (2.20), which is also known as Kingman's approximation:

$$E(QT) = \left(\frac{c_a^2 + c_e^2}{2} \right) \left(\frac{\rho}{1-\rho} \right) E(EPT), \tag{2.20}$$

where ρ is utilization and c_a^2 and c_e^2 stand for SCV of arrival interval and EPT, respectively.

Eq. (2.19a), (2.19b) and (2.20) are the foundations of the approximations for a G/G/1_Run-based non-preemptive product-induced event model. While it is only an approximation for G/G/1 systems, it is exact when the arrival process is Poisson. The exact M/G/1 model can be obtained by substituting c_a^2 with 1, and the exact M/M/1 model can be obtained by substituting both c_a^2 and c_e^2 with 1.

Table 2.2 Parameters for computing QT under the existence of run-based events (by Hopp and Spearman 1996)

Situation	Natural	Preemptive	Non-preemptive
t_e	t_0	t_0 / A	$t_0 + t_p / N_p$
σ_e^2	$t_0^2 c_0^2$	$\sigma_0^2 / A^2 + \frac{\left(m_r^2 + \sigma_r^2\right)(1-A)t_0}{Am_r}$	$\sigma_0^2 + \frac{\sigma_p^2}{N_p} + \frac{N_p - 1}{N_p^2} t_p^2$
c_e^2	c_0^2	$c_0^2 + \left(1 + c_r^2\right) A(1-A) m_r / t_0$	σ_e^2 / t_e^2

Hopp and Spearman (1996) gave an extensive discussion on both run-based preemptive events and run-based non-preemptive product-induced events based on the concept of EPT. By using Eq. (2.20), queueing time (QT) for the above two cases can be

estimated. Table 2.2 summarizes the formulations of EPT and c_e^2 used in Eq. (2.20), where "Preemptive" means run-based preemptive events (2-1-1) and "Non-preemptive" means run-based non-preemptive product-induced events (2-1-2.b).

It can be observed that Kingman's approximation is composed of three elements: variability, utilization and service time. Therefore, mean queueing time can be estimated by two parts: the M/M/1 queue and the variability term, which is approximated by the average of two SCVs. Whitt (1993) conducted numerical experiments on the above approximation with the extensions to multiple servers, and specified the valid ranges of the approximations.

2.4.3 Models for Time-based Preemptive Interruptions

In section 2.4.3 and 2.4.4, we address models for situation (2-2) in Figure 1. Throughout this section, the models and methods we present are closely related to those illustrated by Gaver (1962) and Adan and Resing (2001), but with added insights on the connection between run-based and time-based events, and the linkage between the models and their use in manufacturing.

For time-based interruptions, we analyze the preemptive events (2-2-1) first and start from the simplest case, the M/M/1_Time-based preemptive event model. In this model, the machine can break down anytime instead of only during processing. Jobs arrive according to a Poisson process with rate λ, and service times are exponentially distributed with mean $1/\mu$. The up and down times are also exponentially distributed with means $1/\eta$ and $1/\theta$. For stability, we assume

$$\rho = \lambda / (\mu A) < 1.$$

By PASTA, an arriving job finds on average $E(L^q)$ jobs in queue and sees a working job with probability ρ. Each job encounters on average $\eta(E(L^q)+\rho)/\mu$ breakdowns in total. Furthermore, each arriving job sees the machine is already down

with probability (1-A). The above scenario is almost the same as for the derivation of Eq. (2.6) except for the last term. Therefore,

$$E(QT) = \frac{E(L^q)}{\mu A} + \frac{\rho}{\mu A} + (1-A)\frac{1}{\theta}, \tag{2.21}$$

and applying Little's law and Eq. (2.21), we get

$$E(QT) = \frac{\rho}{1-\rho}\frac{1}{\mu A} + \frac{(1-A)}{1-\rho}\frac{1}{\theta}, \tag{2.22}$$

and

$$E(CT) = E(QT) + E(G) = \frac{1}{1-\rho}\frac{1}{\mu A} + \frac{(1-A)}{1-\rho}\frac{1}{\theta}. \tag{2.23}$$

Comparing Eq. (2.22) with Eq. (2.8), the gap between M/M/1_Time-based preemptive event and M/M/1_Run-based preemptive event models is as follows,

$$\begin{aligned} Gap = &\left(\frac{\rho}{1-\rho}\frac{1}{\mu A} + \frac{1}{1-\rho}\frac{(1-A)}{\theta}\right) \\ &-\left(\frac{\rho}{1-\rho}\frac{1}{\mu A} + \frac{\rho}{1-\rho}\frac{1-A}{\theta}\right) = \frac{(1-A)}{\theta}. \end{aligned} \tag{2.24}$$

This gap is also the gap between M/M/1_Time-based preemptive event model and the preemptive outage model introduced in Factory Physics, since the preemptive outage model is identical to the M/M/1_Run-based preemptive event model.

A more general case is the M/G/1_Time-based preemptive event model. The assumptions are the same as above, except the service time and down time are generally distributed. The first and second moments of the service time are denoted by $E(S)$ and $E(S^2)$. The up time between two breakdowns is exponentially distributed with mean $1/\eta$. The first and second moments of the down time are denoted by $E(D)$ and $E(D^2)$. For stability, we assume

$$\rho_G = \lambda E(G) < 1.$$

Similar to the derivations of Eq. (2.10), an arriving job finds on average $E(L^q)$

jobs in queues, and a working job with probability ρ_G. Furthermore, an arriving job has a certain probability, $(1-A_{NP})(1-\rho_G)$, to see the machine down in a non-processing period. The above scenario is the same as in the derivations of Eq. (2.10) except for the last term. Therefore,

$$E(QT) = E(L^q)E(G) + \rho_G E(R_G) + (1-\rho_G)(1-A_{NP})E(R_D), \tag{2.25}$$

where A_{NP} is availability of the machine during non-processing period, and

$$E(G) = E(S) + E(S)\eta E(D),$$

$$E(G^2) = E(S^2)\left[1+\eta \mathrm{E}(D)\right]^2 + E(S)\eta E(D^2),$$

$$\rho_G = \lambda E(G),$$

$$E(R_G) = E(G^2)/2E(G),$$

$$E(R_D) = E(D^2)/2E(D),$$

$$A_{NP} = \frac{1/(\lambda+\eta)}{1/(\lambda+\eta) + E(D)\eta/(\lambda+\eta)} = \frac{1}{1+E(D)\eta},$$

applying Little's law and Eq. (2.25) we get

$$E(QT) = \frac{\rho_G E(R_G)}{(1-\rho_G)} + \frac{E(D)\eta}{1+E(D)\eta}E(R_D), \tag{2.26}$$

and

$$E(CT) = E(QT) + E(G). \tag{2.27}$$

Both Eq. (2.26) and (2.11) are first proposed by Gaver (1962) in slightly different forms, which considered transfer batch size. The variance of QT has been given by Gaver (1962). Comparing Eq. (2.26) with Eq. (2.11), the gap between M/G/1_Time-based preemptive event models and M/G/1_ Run-based preemptive event models is

$$Gap = (1-A_{NP})E(R_D). \tag{2.28}$$

Property 2 (*Decomposition property of preemptive events*).

For any time-based interruption, there is a gap between its run-based and time-based models. This gap is independent of utilization.

The gap in Eq. (2.24) is a degenerate case of Eq. (2.28) when the down time is exponentially distributed. Eq. (2.28) is very important. Because the models given by Hopp and Spearman (1996) are only applicable to run-based cases, Kingman's approximation of Eq. (2.20) is correct for run-based cases (2-1-1) only. If we apply Kingman's approximation to a time-based preemptive interruption (2-2-1), there will be a systematic error in the approximation. The error takes the form of Eq. (2.28) and is a utilization-independent gap. By taking advantage of the decomposition property, for the time-based cases, we can always calculate their corresponding run-based cases first (by Kingman's approximation), then get the time-based results by adding this gap back. Therefore, for the most general situation (the G/G/1_Time-based preemptive event), a good approximation can be obtained by combining Eq. (2.20) and (2.28).

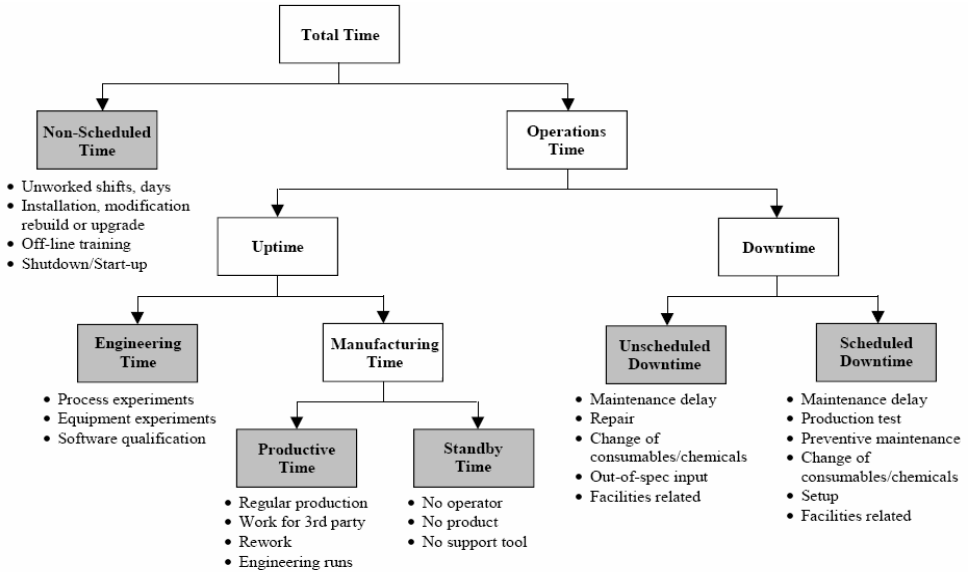

Figure 2.2 Summary of Time (SEMI E10, 2001)

2.4.4 Models for Time-based Non-Preemptive Interruptions

Time-based non-preemptive events (2-2-2) are very common in practice. Actually, most of the events listed in the "Summary of Time" (Figure 2.2) of SEMI E10 are in this category. Process experiments, equipment experiments, preventive maintenance, tool modifications, and change of consumables are examples (or sub-groups) of this type of interruption. The modeling of this type of event becomes extremely complicated if we want to analyze each sub-group individually, since the occurrences of each interruption in a sub-group can have correlations with previous occurrences. The correlation could be based on a period of time, the usage rate (i.e. utilization), or both. For example, a part needs to be replaced every 3 months or after 8000 usages, whichever comes first.

Fortunately, what we usually care about is the overall system performance instead of the behavior of each interruption sub-group. Instead of looking at each specific sub-group, if we look at this category of events from a macroscopic level, the Poisson assumption becomes a reasonable one, since the occurrences of interruptions are triggered by many different sources, and the Palm-Khintchine Theorem may be applied.

The overall behavior of this type of event can be classified as time-based non-preemptive. Furthermore, we usually have some level of control on the occurrence of each interruption. For example, we may postpone the execution of a preventive maintenance (PM) or an equipment experiment until the machine completes its current processing job, or until the machine is idle (without any WIP in queue). The above two scenarios are classified as follows:

Case 1: Postpone a PM to the completion of all jobs in queue,

Case 2: Postpone a PM to the completion of the current job.

In Case 1, the time-based non-preemptive interruptions have low priority and the WIP has high priority. In Case 2, the time-based non-preemptive interruptions have high

priority and the WIP has low priority. Both cases can be modeled by the non-preemptive priority queues with two priorities. The study of non-preemptive priority queues can be traced back to Ashcroft (1950). Cobham (1954) investigates both M/G/1 and M/M/c non-preemptive priority queues and derived their equilibrium expected queueing times. Jaiswal (1968) discusses both preemptive and head-of-the-line (i.e. non-preemptive) priority disciplines in his book, Priority Queues. Although priority queues have long been studied, their application to time-based non-preemptive interruptions described above is new.

We start our analysis from the simplest case, the M/M/1_Non-preemptive priority queue with two priorities. For stability, we assume

$$\rho = (\lambda_1/\mu_1) + (\lambda_2/\mu_2) < 1.$$

The results are as follows (Gross and Harris 1998),

$$E(L_1^q) = \frac{\rho_1^2 / \lambda + \rho_1\rho_2(\mu_1/\mu_2)}{1-\rho_1}, \tag{2.29}$$

$$E(L_2^q) = \frac{\rho_1\lambda_2/\mu_1 + \rho_2/\mu_2}{(1-\rho_1)(1-\rho_1-\rho_2)}, \tag{2.30}$$

$$E(L^q) = E(L_1^q) + E(L_2^q), \tag{2.31}$$

$$E(L_1) = \lambda_1 E(CT_1) = \frac{(\lambda_1\rho_2/\mu_2) + \rho_1}{1-\rho_1},$$

$$E(L_2) = E(L) - E(L_1),$$

$$E(L) = \frac{\rho_1+\rho_2}{1-\rho_1-\rho_2},$$

where

$$\rho_1 = \lambda_1/\mu_1,$$

$$\rho_2 = \lambda_2/\mu_2,$$

λ_1 and λ_2 are the arrival rates of high and low priority jobs, and μ_1 and μ_2 are the service

rates of high and low priority jobs respectively.

If the service times are generally distributed and the discipline is FIFO, the model of M/G/1_Non-preemptive priority queues with two priorities is as follows (Adan and Resing, 2001),

$$E(QT_1) = \frac{\rho_1 E(R_1) + \rho_2 E(R_2)}{1-\rho_1}, \tag{2.32}$$

$$E(QT_2) = \frac{\rho_1 E(R_1) + \rho_2 E(R_2)}{(1-\rho_1-\rho_2)(1-\rho_1)}, \tag{2.33}$$

$$E(QT) = \frac{\lambda_1}{\lambda_1+\lambda_2} E(QT_1) + \frac{\lambda_2}{\lambda_1+\lambda_2} E(QT_2), \tag{2.34}$$

$$E(CT_i) = E(QT_i) + E(S_i), \quad i = 1, 2$$

$$E(CT) = E(QT) + \frac{\lambda_1}{\lambda_1+\lambda_2} E(S_1) + \frac{\lambda_2}{\lambda_1+\lambda_2} E(S_2),$$

where

$$\rho_1 = \lambda_1 E(S_1) \text{ and } \rho_2 = \lambda_2 E(S_2),$$

$$E(R_i) = E(S_i^2)/2E(S_i), \quad i = 1, 2$$

λ_1 and λ_2 are the arrival rates of high and low priority jobs, respectively. For stability, we assume

$$\rho = \lambda_1 E(S_1) + \lambda_2 E(S_2) < 1. \tag{2.35}$$

In practice, the control mechanism may be more complex than the above assumptions. For example, we may postpone the occurrence of a PM to some period when the machine is less busy, but may not postpone it too much without jeopardizing the quality of products. In this case, we may model it as follows: The interruption has low priority until its queueing time reaches a predetermined threshold, after which it switches to high priority. We call this scenario “delayed priority queues”. However, its derivations involve the analysis of high dimensional Markov chains and will not be discussed here. It

is left as a direction for future research.

Although the exact model for the delayed priority queue is not available, we know its upper and lower bounds. The queueing time of a PM is between the queueing time of the high priority job in Case 2 (lower bound), and the queueing time of the low priority job in Case 1 (upper bound). One thing we need to keep in mind is that the WIP and utilization of real manufacturing systems fluctuate every day. Therefore, in practice, the situation could be closer to Case 1 when the predicted machine utilization is low in the coming few days, and closer to Case 2 when the predicted machine utilization is high in the coming few days. When the predicted utilization is low, we have chances to perform the PM without interrupting production runs too much. On the other hand, if the predicted utilization is high in the coming few days, there is no motivation to wait and take the risk of sacrificing product quality, since we know the PM will interfere with production and increase system variability anyway. In this case, it is best to do it as soon as possible.

2.5 Integration of Models

In Section 2.4, we described how to model different categories interruptions. However, in practice, events of all categories can occur at the same machine. An integrated model can describe the overall system performance under the impact of all kinds of interruptions.

Similar to what we have done before, we will conduct mean value analysis to get the queueing time of the integrated model. Before proceeding to the detailed mean value analysis, we want to have some observations on generalized service time first. Based on Eq. (2.2), the product-induced setup time and all the preemptive interruptions that occur during processing have been considered in GST. The preemptive interruptions during processing can be either run-based or time-based. While the run-based interruptions are fully covered by GST, GST only covers part of the time-based interruptions, since some

of the time-based interruptions may occur during non-processing time. Therefore, the generalized service time contains the original service time, run-based preemptive events (2-1-1), run-based non-preemptive product-induced events (2-1-2.b), and some of the time-based preemptive events (2-2-1), i.e.

$$G_1 = S_1 + \sum_{i=1}^{N_r(S_1)} D_{r\_i} + \sum_{i=1}^{N_t(S_1)} D_{t\_i} + T_p, \tag{2.36}$$

where G_1 stands for generalized service time, S_1 stands for service time, $N_r(S_1)$ is the number of run-based preemptive events during S_1, D_r is the length of the run-based preemptive events, $N_t(S_1)$ is the number of time-based preemptive events (e.g. breakdowns) during S_1, D_t is the length of the time-based preemptive events, and T_p stands for the duration of a run-based non-preemptive product-induced event experienced by a job.

The first and second moments of generalized service time in the integrated models are given as follows. Derivations are provided in Appendix A.

$$E(G_1) = E(S_1) + E(S_1)\eta_r E(D_r) + E(S_1)\eta_t E(D_t) + E(T_p), \tag{2.37}$$

$$\begin{aligned} E(G_1^2) = E(S_1^2)&\left\{\begin{matrix}\left[1+\eta_r \mathrm{E}(D_r)\right]^2 + \left[1+\eta_t \mathrm{E}(D_t)\right]^2 \\ +2\mathrm{E}(T_p) - 1\end{matrix}\right\} \\ &+ E(S_1)\eta_r\left[E(D_r^2) + 2\mathrm{E}(T_p)\mathrm{E}(D_r)\right] \\ &+ E(S_1)\eta_t\left[E(D_t^2) + 2\mathrm{E}(T_p)\mathrm{E}(D_t)\right] + \mathrm{E}(T_p^2). \end{aligned} \tag{2.38}$$

There are two possible cases for the time-based non-preemptive events (2-2-2). Each case has its own assumptions: The WIP has high priority (relative to the time-based non-preemptive interruptions) in Case 1 and low priority in Case 2. Therefore, we will also have two different integrated models derived based on the assumptions given in Case 1 and Case 2.

Notation:

S_1: Service time of a job

λ_1: Arrival rate of a job

G_1: Generalized service time of a job

$\rho_1 = \lambda_1 E(G_1)$

QT_1: Queueing time of a job

CT_1: Cycle time of a job

S_2: Duration of a time-based non-preemptive event

λ_2: Arrival rate of a time-based non-preemptive event

G_2: General recovery time of a time-based non-preemptive event

$\rho_2 = \lambda_2 E(G_2)$

D_r: Duration of a run-based preemptive event

η_r: Arrival rate of a run-based preemptive event

D_t: Duration of a time-based preemptive event

η_t: Arrival rate of a time-based preemptive event

T_s: Duration of a run-based non-preemptive state-induced event

T_p: Duration of a run-based non-preemptive product-induced event experienced by a job

P: Duration of a run-based non-preemptive product-induced event

N_p: Average number of jobs processed by a machine between two consecutive setups

Analysis for Case 1:

In Case 1, we will postpone a PM to the completion of all jobs in queue. The time-based non-preemptive event has low priority, while the job has high priority.

By PASTA, an arriving job finds on average $E(L_1^q)$ jobs in queues, and sees the machine occupied by a job (with probability ρ_1), occupied by a time-based non-preemptive event (with probability ρ_2), or none of the above (with probability $(1-\rho_1-\rho_2)$).

If an arriving job finds the machine is not occupied by a job or a time-based non-preemptive event, it must find the machine in one of the three states: up, doing a state-induced setup, or handling a time-based preemptive event.

If the machine is up, the arriving job will cause a state-induced setup before processing starts. If the job arrives when the machine is in a state-induced setup, it must wait until the remaining setup process is completed. If the job arrives when the machine is in a time-based preemptive event, it must wait until the machine is recovered plus a state-induced setup time. Therefore,

$$\begin{aligned} E(QT_1) &= E(L_1^q)E(G_1) + \rho_1 E(R_{G1}) + \rho_2 E(R_{G2}) \\ &+ (1-\rho)\left[\begin{array}{l} \dfrac{1/(\lambda_1+\eta_t)}{\Sigma}E(T_s) + \dfrac{E(T_s)}{\Sigma}E(R_{Ts}) \\ + \dfrac{\left(\dfrac{E(D_t)\eta_t}{(\lambda_1+\eta_t)}\right)}{\Sigma}\left(E(R_{Dt}) + E(T_s)\right) \end{array}\right]. \end{aligned} \tag{2.39}$$

Combining Little's law and Eq. (2.39) we get

$$\begin{aligned} E(QT_1) &= \frac{\rho_1}{1-\rho_1}E(R_{G1}) + \frac{\rho_2}{1-\rho_1}E(R_{G2}) \\ &+ \frac{(1-\rho)}{1-\rho_1}\left[\begin{array}{l} \dfrac{1/(\lambda_1+\eta_t)}{\Sigma}E(T_s) + \dfrac{E(T_s)}{\Sigma}E(R_{Ts}) \\ + \dfrac{\left(\dfrac{E(D_t)\eta_t}{(\lambda_1+\eta_t)}\right)}{\Sigma}\left(E(R_{Dt}) + E(T_s)\right) \end{array}\right], \end{aligned} \tag{2.40}$$

and

$$E(CT_1) = E(QT_1) + E(G_1),$$

where

$$\Sigma = 1/(\lambda_1+\eta_t) + E(D_t)\eta_t/(\lambda_1+\eta_t) + E(T_s),$$

$$\rho = \rho_1 + \rho_2,$$

$$E(R_{Gi}) = E(G_i^2)/2E(G_i), \qquad i = 1,\ 2$$

$$E(G_2) = E(S_2), \quad E(G_2^2) = E(S_2^2),$$

$$E(G_1) \textit{ is given by Eq. } (2.37),$$

$$E(G_1^2) \textit{ is given by Eq. } (2.38),$$

$$E(T_p) = E(P) / N_p,$$

$$E(R_{Ts}) = E(T_s^{\,2}) / 2E(T_s),$$

$$E(R_{Dt}) = E(D_t^{\,2}) / 2E(D_t).$$

Analysis for Case 2:

In Case 2, we will postpone a PM to the completion of the current job. The time-based non-preemptive event has high priority, while the job has low priority. Since the jobs have low priority, in order to get the average queueing time of a job, we have to calculate the queueing time of a PM first.

By PASTA, a new PM finds on average $E(L_2^q)$ PMs in queues, and sees that the machine is occupied by a job (with probability ρ_1), occupied by a time-based non-preemptive event (with probability ρ_2), or none of the above (with probability $(1-\rho_1-\rho_2)$). If a new PM finds the machine not occupied by a job or a time-based non-preemptive event, it must find the machine in one of the following three states: up, during a state-induced setup, or during a time-based preemptive event.

If the machine is up, the new PM will have no queueing time. If the PM arrives when the machine is in a state-induced setup, it must wait until the remaining setup process is complete. If the job arrives when the machine is in a time-based preemptive event, it must wait until the machine is available. Therefore,

$$E(QT_2) = E(L_2^q)E(G_2) + \rho_1 E(R_{G1}) + \rho_2 E(R_{G2}) + (1-\rho)\left[\frac{E(T_s)}{\Sigma}E(R_{Ts}) + \frac{\left(\frac{E(D_t)\eta_t}{(\lambda_1+\eta_t)}\right)}{\Sigma}E(R_{Dt})\right]. \quad (2.41)$$

Applying Little's law to Eq. (2.41) we get

$$E(QT_2) = \frac{\rho_1}{1-\rho_2}E(R_{G1}) + \frac{\rho_2}{1-\rho_2}E(R_{G2}) + \frac{(1-\rho)}{1-\rho_2}\left[\frac{E(T_s)}{\Sigma}E(R_{Ts}) + \frac{\left(\frac{E(D_t)\eta_t}{(\lambda_1+\eta_t)}\right)}{\Sigma}E(R_{Dt})\right]. \quad (2.42)$$

Similar to the derivations given for Eq. (2.39), the queueing time of a job can be derived accordingly. By PASTA, an arriving low priority job finds on average $E(L_1^q)$ jobs and $E(L_2^q)$ time-based non-preemptive events in queues, and sees the machine occupied by a job (with probability ρ_1), occupied by a time-based non-preemptive event (with probability ρ_2), or none of the above (with probability $(1-\rho_1-\rho_2)$). If an arriving job finds the machine not occupied by a job or a time-based non-preemptive event, it must find the machine in one of the following three states: up, during a state-induced setup, or during a time-based preemptive event. Therefore,

$$E(QT_1) = E(L_1^q)E(G_1) + E(L_2^q)E(G_2) + \rho_1 E(R_{G1}) + \rho_2 E(R_{G2}) + (1-\rho)\left[\frac{1/(\lambda_1+\eta_t)}{\Sigma}E(T_s) + \frac{E(T_s)}{\Sigma}E(R_{Ts}) + \frac{\left(\frac{E(D_t)\eta_t}{(\lambda_1+\eta_t)}\right)}{\Sigma}\left(E(R_{Dt}) + E(T_s)\right)\right]. \quad (2.43)$$

Applying Little's law to Eq. (2.43) we get

$$E(QT_1) = \frac{\rho_1}{1-\rho_1} E(R_{G1}) + \frac{\rho_2}{1-\rho_1}\left[E(QT_2) + E(R_{G2})\right]$$
$$+ \frac{(1-\rho)}{1-\rho_1}\left[\frac{1/(\lambda_1+\eta_t)}{\Sigma} E(T_s) + \frac{E(T_s)}{\Sigma} E(R_{Ts}) + \frac{\left(\frac{E(D_t)\eta_t}{(\lambda_1+\eta_t)}\right)}{\Sigma}\left(E(R_{Dt}) + E(T_s)\right)\right], \quad (2.44)$$

and

$$E(CT_1) = E(QT_1) + E(G_1),$$

where

$$\Sigma = 1/(\lambda_1+\eta_t) + E(D_t)\eta_t/(\lambda_1+\eta_t) + E(T_s),$$

$$\rho = \rho_1 + \rho_2,$$

$$E(R_{Gi}) = E(G_i^2)/2E(G_i), \qquad i = 1,\ 2$$

$$E(G_2) = E(S_2),$$

$$E(G_2^2) = E(S_2^2),$$

$E(G_1)$ *is given by Eq.* (2.37),

$E(G_1^2)$ *is given by Eq.* (2.38),

$$E(T_p) = E(P)/N_p,$$

$$E(R_{Ts}) = E(T_s^2)/2E(T_s),$$

$$E(R_{Dt}) = E(D_t^2)/2E(D_t).$$

As with the conclusions given in the analysis of time-based non-preemptive events (2-2-2), depending on the PM scheduling policy, the queueing time of a job in practice could be the values given in Eq. (2.40), (2.44) or any value in between. Eq. (2.40) gives a lower bound and Eq. (2.44) gives an upper bound of the true queueing time.

In the above two models, we assume a preemptive event will only preempt the processing of a job instead of another interruption. For example, a power outage (2-2-1)

will not occur during an out-of-spec input (2-1-1), a setup (2-1-2), a PM (2-2-2), or another time-based preemptive event (2-2-1). We need to keep in mind that Eq. (2.38) would become an approximation if the preemptive events could preempt an interruption. After all, we can not prevent a power outage from occurring during a setup or a PM.

2.6 Resource Contention Problems

Based on the level of controllability, we may model interruptions by two different approaches: (a) downtime events (ample resource), or (b) resource contention. Until now, all the queueing models we have introduced in previous sections are downtime event models, which assume the resources are sufficient. In this section, we are going to look into a more complicated situation, where the resources are limited. If we limit our attention to only time-based preemptive interruptions, this type of problem, also called machine interference or machine repairman problem, has been studied since the 1930s (Khintchine 1933). Jaiswal (1968) analyzes the M/G/1/N finite source model. Stecke and Aronson (1985) and Haque and Armstrong (2007) give comprehensive reviews on this topic. For the comparison with downtime event models, we give a brief introduction to resource contention models.

Before going into the detail, the first observation is that there are different kinds of interruptions on the shop floor, not only machine breakdowns. Some may be handled by repairmen, but some may need operators (e.g. setup) or engineers (e.g. experiments). This is one of the reasons that we call it a resource contention problem instead of a machine repairman problem, since the former is more general. The recovery from almost of all interruptions involves some scarce resource. Those resources can be operators, engineers, mask sets, support tools or parts. All the models presented in Section 2.4 and 2.5 share a common implicit assumption: the resources needed to recover from interruptions are always available. If those resources are shared among multiple machines

and we do not have full control on the occurrences of interruptions, then resource contention also should be considered in modeling behavior.

When the resources are always available or need not be shared with others, those events can be simply modeled as downtime events. For example, equipment experiments are properly modeled as downtime events, since we can choose not to start the experiment until all resources, such as engineers and parts, are ready. Product-induced setups have a similar property, since we can choose to run the same products until the operator is available. Therefore, in some situations, downtime event models are adequate for the needs of modeling a non-preemptive interruption. However, for a preemptive interruption, the downtime event models may not be adequate. We may have to resort to resource contention models.

An important observation is that not all interruptions with limited recovering resources (e.g. repairmen) need to be modeled by machine repairman models. For example, even though there is only one repairman tending multiple machines, if the occurrence of interruptions is controllable, downtime event models may suffice. The key is if there is resource contention or not. This is another reason we call this type of problems resource contention problems.

Engineers, operators, over-head transportation systems and support tools are resources that typically are shared. If the resource is scarce and the interruption is time-based and preemptive, this type of event can be modeled using finite source queues. In this type of model, we assume the calling population is finite, and future event occurrence probabilities are functions of system state. If the time a calling unit spends outside the system is exponential with mean $1/\lambda$, service times are exponentially distributed with mean $1/\mu$, and there is only one repairman, the model can be modeled by the M/M/1 finite source queues as follows (Palm 1958 and Gross and Harris 1998),

$$P_0 = \left(\sum_{n=0}^{M} \frac{M!}{(M-n)!} \left(\frac{\lambda}{\mu} \right)^n \right)^{-1}, \tag{2.45}$$

$$P_n = \frac{M!}{(M-n)!} \left(\frac{\lambda}{\mu} \right)^n P_0, \tag{2.46}$$

$$E(L) = \sum_{n=0}^{M} n P_n, \tag{2.47}$$

$$E(L^q) = L - \frac{\lambda}{\mu}(M - L), \tag{2.48}$$

$$E(CT) = \frac{E(L)}{\lambda(M - E(L))}, \tag{2.49}$$

$$E(QT) = \frac{E(L_q)}{\lambda(M - E(L))}, \tag{2.50}$$

where M is the size of the calling population (or the number of machines). Since the breakdown is preemptive and time-based in the above model, based on the classification in Figure 2.1, it is indeed a time-based preemptive resource contention model (2-2-1).

For the case of multiple repairmen, Palm (1958) and Gross and Harris (1998) provide results for the more general model, M/M/c finite source queues. Peck and Hazelwood (1958) offer a book of tables for the results of M/M/c finite source queues. Wang and Sivazlian (1990) obtain G/G/c finite source approximate models through diffusion approximation. Gupta and Rao (1994 and 1996) develop a recursive method for the M/G/1 finite source queue. They also extend the previous results by considering the spares, which means a failed machine can be immediately replaced by a spare machine, if spares are available.

In addition to the repair time, many previous researchers also took into account the patrolling time. Mark, Murphy and Webb (1957) first consider constant walking time in a single operator model, where the model has constant service times and random

breakdowns. Reynolds (1975) analyzes M/M/c finite source queues by considering walking times based on the shortest distance priority queue discipline. This type of extension further complicates the resource contention models and it is important when walking time is significant in the process. However, compared with the repair times, walking times sometimes are short enough to be negligible in practice.

Although downtime event models are adequate for some non-preemptive interruptions, we still need resource contention models in many situations if we want to model the situations with high fidelity. In practice, the models of resource contention problems are much more involved than the downtime event models. For example, a repairman may tend multiple workstations, where each workstation may have several repair time distributions caused by different types of interruptions. Therefore, it would be nice if we can approximate the performance of a resource contention problem using downtime event models.

Morrison et al. (2007) treat the resource contention problem as a special case of the run-based preemptive downtime events. They assume the durations caused by resource contention are independent of production time, and are generally distributed with mean m_I and standard deviation s_I. They call this case “idle with work available” and modeled it as a run-based preemptive event.

The flaws of this approach are obvious. It assumes the resource is always sufficient. Furthermore, it fails to distinguish the differences between a run-based and a time-based event, since some resource contention problems can be time-based, such as a power outage. It also fails to distinguish the differences between a preemptive and non-preemptive event, since a resource contention problem could be either preemptive or non-preemptive. Comparing their approximate model with the simulation results, the errors are reporter to be from 1.4% to 21.3%, depending on the machine types and utilization.

The above approach can be easily improved by our knowledge of the interruption classifications. Instead of treating all resource contention problems as run-based

preemptive downtime events, we can use the corresponding downtime event model to approximate the performance of a resource contention problem by assuming the resource is sufficient or the impact from resource contention can be incorporated into the downtime distributions. This simplification can greatly reduce the model complexity.

Although resource contention problems have long been studied, previous researchers mainly focus on the property of resources (such as spares, repairmen dispatch and patrolling times, etc.) instead of the nature of interruptions. With the knowledge of interruption classifications, we know that the interruptions which cause resource contention can be attributed to many different reasons. The derivations of other necessary models (for different types of interruptions) and the integrated models with considering the existence of resource contention are left for future research.

In addition to the cases discussed above, the resources may also share some other limited resources. For example, to fix a certain type of breakdowns, a repairman may need a special tool for some critical steps during the whole recovery process. The factory may only have one or two tool sets due to cost concern. That means a group of machines share a few repairmen, who share few tool sets. The repairmen may also share the overlapped work space, if the machines are located close to each other, or they may share the same water fountain, if they feel thirsty in the recovery process. In practice, the situation is much more complicated than the model. It is almost impossible to consider all the details. Capturing the key factors which dominate the outcome is important.

2.7 Simulation Experiments

The thesis of this chapter is that the right model should be used for each specific situation in which interruptions occur. Errors will be introduced if improper models are used, but the reader may ask how big the errors can be.

Actually, the answer for the M/G/1 based models is given in Eq. (2.28), which is

the gap between M/G/1_Time-based preemptive event models and M/G/1_Run-based preemptive event models. Nevertheless, all the results of this chapter until now are based on mathematical derivations. In order to give practitioners better perception of the concepts behind the mathematical models, we evaluate some basic models by simulation and gain some insights from it.

The simulation studies were conducted using the software eM-Plant®. We conduct the demonstration on the M/M/1_Time-base preemptive events systems (2-2-1). A single non-tandem server with exponential service time is assumed. The mean service time is 30 min. Intervals of lot release, failures, and repairs are all exponentially distributed. A total of 9 experiments are conducted for 9 different input rates. The mean service time, MTBF, and MTTR are 30, 1020, and 180 minutes, respectively. The availability is therefore 85% (= 1 – 180/1200).

We first want to look at the relationship between utilization and effective process time. The warm up period of each experiment is 10 years. For each input rate, 40,000 samples (lots) are collected after the warm up period. Because all interruptions are time-based, EPT varies with utilizations as illustrated in Figure 2.3.

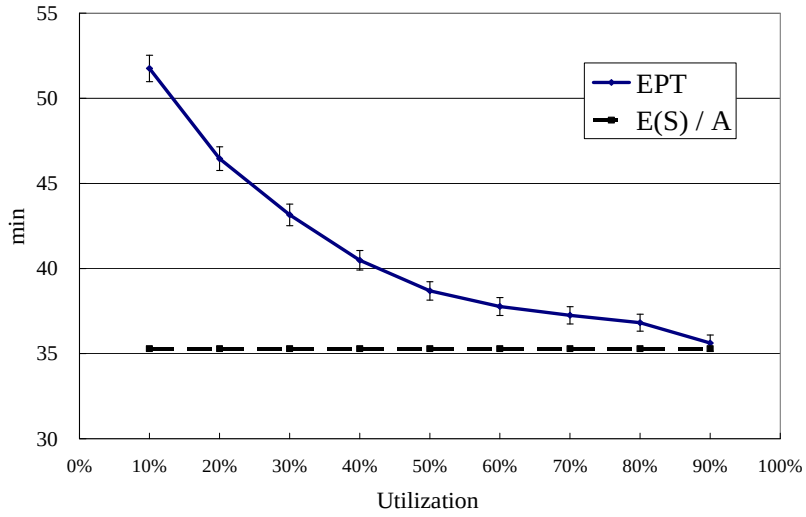

Figure 2.3 Effective process times with time-based events

When there is no time-based event, effective process time can be calculated by Eq. (2.3c). The value is 35.3 min (= 30/0.85). The true effective process time approaches 35.3 in heavy traffic (Wu et al., 2008), which means the errors caused by applying the wrong model (i.e. run-based preemptive model) to the time-based preemptive interruptions diminishes in heavy traffic. This observation is consistent with the decomposition principle of time-based events. Since the gap is independent of utilization, but the queueing time increases with utilization increasing, the error percentage decreases with utilization increasing.

What will be the results if we apply Eq. (2.20) to the systems with time-based interruptions? To answer this question, a second set of simulations have been conducted. The simulation parameters are basically the same as the initial experiments. However, in order to have smaller confidence intervals on cycle times, the warm up period has been increased to 50 years (ten years are not sufficient to get a small confidence interval in this case). For each input rate, 100 replications are collected. Each replication is composed of 200,000 samples (lots). Table 2.3 gives a comparison of cycle times among the results from simulation, Eq. (2.20) and (2.23).

Table 2.3 Comparison among different cycle times

Unit: min					Diff %		Gap		
Util.	*SCT*		*TCT*	*FCT*	*TCT*	*FCT*	*TCT-SCT*	*FCT-SCT*	*FCT-TCT*
10%	69.23	± 0.07	69.22	42.22	0.0%	-39.0%	-0.01	-27.01	-27.00
20%	77.94	± 0.11	77.87	50.87	-0.1%	-34.7%	-0.07	-27.07	-27.00
30%	88.89	± 0.15	88.99	61.99	0.1%	-30.3%	0.10	-26.90	-27.00
40%	103.70	± 0.23	103.82	76.82	0.1%	-25.9%	0.13	-26.87	-27.00
50%	124.76	± 0.29	124.59	97.59	-0.1%	-21.8%	-0.17	-27.17	-27.00
60%	155.70	± 0.47	155.74	128.74	0.0%	-17.3%	0.04	-26.96	-27.00
70%	207.50	± 0.74	207.65	180.65	0.1%	-12.9%	0.15	-26.85	-27.00
80%	312.72	± 1.68	311.47	284.47	-0.4%	-9.0%	-1.25	-28.25	-27.00
90%	624.25	± 6.60	622.94	595.94	-0.2%	-4.5%	-1.31	-28.31	-27.00

In Table 2.3, SCT is the simulated cycle time, TCT is the theoretical cycle time from Eq. (2.23), and FCT is the approximated cycle time from Eq. (2.20). At all utilizations, FCT is a very good approximation, much superior to TCT. In fact, TCT performs quite poorly except for very high utilizations. These data reinforce the importance of using the model appropriate to the types of interruptions.

It should be noted that, in the last column of Table 2.3, the gap between FCT and TCT is a constant as explained in Section 2.4. This value can be verified by Eq. (2.24).

2.8 Comparison of Queueing Classifications and SEMI E10

The purpose of SEMI E10 is to "establish a common basis for communication between users and suppliers of semiconductor manufacturing equipment by providing standards for measuring RAM performance of that equipment in a manufacturing environment," where RAM stands for "reliability, availability, and maintainability". The classification of time proposed in SEMI E10 is summarized in Figure 2.2. The purposes of these two types of classifications (Figure 2.1 and Figure 2.2) are very different. However, they both attempt to classify all the shop floor events.

The activity classifications of SEMI E10 are commonly adopted for productivity improvement projects in practice. By comparing these two classifications, we can see how each queueing model interacts with the activities in practice from the view point of productivity improvement.

Un-worked shifts, installation, modification, rebuild or upgrade, off-line training, shutdown/start-up, which belong to Non-Schedule Time are time-based non-preemptive events. Process and equipment experiments, software qualification listed under Engineering Time are also time-based non-preemptive events.

Activities under Productive Time, such as regular production, work for third party, rework, and engineering runs can be viewed as product mix variability. However, a

portion of these activities also belongs to run-based non-preemptive state-induced events, since load and unload are classified into Productive Time based SEMI E10.

Under Unscheduled Downtime, while maintenance delay is a resource contention problem, change of consumables/chemicals, repair, and facilities related unscheduled downtime (e.g., power outage, or facilities related, etc.) are time-based preemptive events in general. Out-of-spec input can be viewed as a run-based preemptive event. It should be noted that some facilities related unscheduled downtime events can be non-preemptive, such as environmental issues (e.g. particle count, humidity, vibration and temperature). However, the majority of unscheduled downtime events are preemptive in nature, since unscheduled downtime implies a not controllable event. The bottom line is that if the unscheduled downtime events have direct impacts on machines' processing, such as power outage, they are preemptive; most unscheduled downtime events are preemptive. However, for those unscheduled downtime events which are non-preemptive, the queueing models presented previously have to be modified, since the availability in the previous models is defined as

$$A = \frac{m_f}{m_f + m_r}, \tag{2.4}$$

where m_f is defined as

$$MTBF = \frac{productive\ time}{\#\ of\ failures\ that\ occur\ during\ productive\ time}, \tag{2.51}$$

and failure is defined as "any unscheduled downtime event that changes the equipment to a condition where it cannot perform its intended function." If failures can be non-preemptive, the results of Eq. (2.51) will be misleading. For example, suppose the first unplanned interruption occurs at 1:00, and the remaining processing time is 3 hours. The second unplanned interruption, then, occurs at 2:00. Although the above events cannot be modeled by preemptive event models, those non-preemptive unscheduled events can be modeled by the non-preemptive models (2-1-2 or 2-2-2) as we explained in Section 2.4.2

and 2.4.4.

In the category of Scheduled Downtime, production test, preventive maintenance, change of consumable/chemicals, and facilities related scheduled downtime all are time-based non-preemptive events. However, maintenance delay is a resource contention problem. Setup is a run-based non-preemptive event.

Under Standby Time, “no product” does not belong to any category in Figure 1, but simply corresponds to the idle time of a machine. The situations with “no operator” and “no support tool” are resource contentions, but do not belong to any of the categories in Figure 2.1. They can only affect a machine through the occurrence of the other events, and their impact on the system must be modeled together with those events. In the other words, their impacts on a machine are indirect. For example, a machine may wait for support tools during a process experiment or change of consumables, and a machine may wait for operators during a setup. Therefore, in SEMI E10 Summary of Time, the standby time caused by no operator and no support tool will occur together with one of the above events in other categories.

Through the above analysis, we conclude that the majority of the events on the shop floor are time-based, and most of the time-based events are non-preemptive. Because the purpose of SEMI E10 is to measure equipment performance, understanding the sources of the events is essential. However, the SEMI E10 classification is not designed for the needs of applying queueing models. For the purpose of applying queueing theory, the classification we propose in this chapter will avoid that confusion.

2.9 Conclusion

We have proposed a comprehensive classification of interruptions in practical manufacturing systems. Compared with the results from previous research, this new classification gives a more powerful tool to understand the behavior of practical

manufacturing systems. Through the classification, we explain the way to model the impact of different types of interruptions on manufacturing process performance by queueing theory. Exploring the underlying structure by classifying events related to machine breakdowns and the accompanying insights is the first step towards this goal. Insightful classification not only helps us understand the property of each category better, but also offers an overview on the relations of all types of events, which leads to the clear definition of GST.

One of the fundamental goals of this work is to create a clearer understanding of the capabilities of theoretical (queueing) models to estimate the performance of real manufacturing systems. Achieving this goal requires precision in defining terms that are used in each domain, e.g., “service time”. Failure to properly define common terms can lead to misapplication of queueing models. Achieving the goal also requires precision in describing the behavior of manufacturing systems. Imprecise characterization may lead to misapplication of queueing models, or may obscure opportunities to apply them.

Achieving greater precision in the language of queueing and the language of manufacturing will also reveal new opportunities for analysis, by revealing structure in manufacturing behavior that may be exploited to adapt existing models or develop new models.

One of the important findings is the cycle time gap between time-based and run-based preemptive models. It is the product of unavailability of a machine during non-processing period and the residual downtime. This gap is independent of machine utilization, and offers us a way to approximate the performance of a time-based G/G/1 system. Because of the classification, time-based non-preemptive events can be dealt with through the assumption of Poisson arrival processes in Section 2.4.4. In Section 2.5, we have also presented integrated models to predict the performances of an M/G/1 system under the impacts of all kinds of events.

We should pay special attention to resource contention problems. This type of

interruption is often confused with downtime events. For example, in Figure 2.2, “Summary of Time”, “no operator” is listed together with “maintenance delay” under different categories of the total time. But we know there could be also a “no repairman” state during “maintenance delay”. However, due to the different purpose of classifications, it is just covered by “maintenance delay” without further classification. From the viewpoint of queueing theory, resource contention events should not be mixed up with downtime events. Their models are different.

Although we have demonstrated that all the shop floor activities in Figure 2.2 can be classified by this new classification, there is still a possibility that some events have not been considered. For example, Model C, “breakdowns occur homogeneous in time, repair process initiated if customers are present at station”, introduced by Avi-Itzhak (1963), can be viewed as a variation of the time-based preemptive downtime events. However, the chance of its occurrence in a semiconductor fab will be small, since machines are highly automated and the majority of the time-based breakdowns can be recognized (by alarms or messages) at the beginning of the failures.

Instead of claiming the classifications presented in this chapter are ultimate and complete, we would rather treat it as a step in applying queueing theory to manufacturing systems in a more systematic way. Further improvements on the classifications and more precise queueing models to describe the behaviors of manufacturing systems are expected and left for future research.

CHAPTER 3

PARALLEL BATCH PROCESSING

Essentially, all models are wrong, but some are useful. ~ George E. P. Box

3.1 Introduction

There are two main types of batches, process batches and transfer batches, and both play important roles in manufacturing systems. Generally, a process batch is defined as processing a pre-determined group of jobs in a certain period of time without being interrupted by other product groups. The job group has no specific composition and can be composed of a single product or multiple products, as long as they use the same recipe (for a parallel batch machine) or do not induce setup (for a serial batch machine). The number of the pre-determined group of jobs, i.e. batch size, is constrained by the process maximum batch size (if any). The machine could process all jobs together, or process all jobs of this product group one by one consecutively without being interrupted by other product groups. We say a machine is a batching machine if its maximum process batch size is greater than one.

We call a process batch a serial batch if products are processed one by one consecutively without being interrupted by other product groups. Based on Hopp and Spearman (1996), the serial batch size is "the number of jobs of a common family processed before the workstation is changed over to another family." A serial batch can be composed of a single product or multiple products, as long as they do not induce setup. The setup induced by a serial batch is called a product-induced setup. A larger average serial batch size will reduce the setup frequency, but may potentially sacrifice customer due date performance.

We call a process batch a parallel batch if all jobs are processed together at the

same time. Furnaces and ovens are typical examples of parallel batch machines. The parallel batch size is usually determined by the physical capacity of a machine, but it could also be determined by the process constraints. For example, the common physical capacity of a furnace in a 300mm semiconductor fab is four lots. However, a furnace parallel batch capacity is sometimes reduced to three due to process quality concerns. A parallel batch can be composed of a single product or multiple products, as long as they use the same recipe at this batching machine.

In addition to process batches, transfer batches are also commonly seen in production lines. Hopp and Spearman (1996) define the transfer batch as "the number of parts that accumulate before being transferred to the next station." In general, transfer batches are caused by the mismatch between transfer units and process units. Therefore, the definition of transfer batch is highly dependent on the process unit of a machine. If the process of a machine is lot based, the transfer batch size is the number of lots. If the process of a machine is wafer based, the transfer batch size is a multiple of wafers. For example, if each lot consists of 24 wafers, when lots arrive one (lot) by one (lot) to a wafer based CVD tool, the transfer batch size is 24 wafers (in one lot). However, since the process of a wet bench is lot based, if lots arrive one by one, the transfer batch size is one lot for a wet bench.

Since the batching process is a potential cause of long cycle time, modeling batch processing correctly is critical in understanding the performance of practical manufacturing systems. Batch processing has captured researchers' attention for a long time and been rigorously studied. The first paper on this topic may be traced back to Bailey (1954), who modeled a simple queueing process in which customers arrive at a single queue at random, and are served in a batch with a fixed maximum batch size.

In 1983, Chaudhry and Templeton summarized the state of the art up to that time in their book, A First Course in Bulk Queues. Two types of batch processing were addressed: bulk-arrival queues and bulk-service queues: bulk arrivals correspond to

transfer batches and bulk services correspond to parallel process batches. Because they focused on the queueing models which can be solved exactly, they did not address G/G/1 based approximate models. Furthermore, models of serial process batches were not discussed.

In the late 1980s, the G/G/1 based approximate models for $G/G^k/1$ queues were proposed by Bitran and Tirupati (1989a) and Segal and Whitt (1989). Their approximation is based on the decomposition of the cycle time into three parts: wait-to-batch time, queueing time and service time, where the queueing time is obtained by G/G/1 approximations. This approximate model is generally applied to understand the behavior of parallel batches in practical manufacturing systems.

Hopp and Spearman (1996) summarized previous work and introduced models for parallel process batches, serial process batches and transfer batches. However, their serial process batch models are limited to batch arrivals where the transfer batch size is the same as the serial batch size.

The approximate parallel batch model proposed by Bitran and Tirupati (1989a) offers us a flexible and powerful tool for describing the behavior of parallel batching machines. However, a potential issue with this approach is the assumption of independence between wait-to-batch time and queueing time, which is not satisfied in general.

By carefully examining this issue, we develop an improved model to approximate the performance of parallel batching machines using the analytical solution from the $M/M^k/1$ model. We also adopt the decomposition approach, and we make our decomposed model be the same as the $M/M^k/1$ model when the arrival process is Poisson and the service time is exponential. The results are then further generalized to the $G/G^k/1$ queues. By doing this, the new decomposition approximation yields the exact solution at the $M/M^k/1$ case and gives smaller errors at the $G/G^k/1$ cases.

3.2 The Analysis

The G/G/1 based approximate models for parallel process batches decompose the cycle time into three parts: wait-to-batch time (WTBT), queueing time (QT) and service time (ST). The structure is illustrated in Figure 3.1. The intention behind this decomposition is to approximate the duration of the three time segments. To guarantee the success of this decomposition, we need to make sure that each segment performs independently of the others.

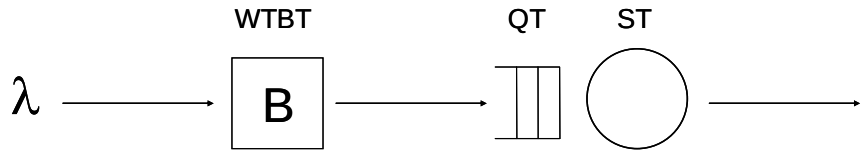

Figure 3.1 The structure of parallel process batches.

Based on this scheme, if the job inter-arrival times are independent and identically distributed, Hopp and Spearman (1996) propose the following model to approximate the average cycle time of parallel process batches,

$$CT \cong \frac{k-1}{2\lambda} + \left(\frac{c_a^2/k + c_b^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu} + \frac{1}{\mu}, \qquad (3.1)$$

where

$$\rho = \lambda / k\mu,$$

and k is the fixed parallel batch size. Therefore, each batch contains k jobs. λ is the arrival rate of jobs (jobs/hour), μ is the service rate of the batching machine (batches/hour), c_a is the coefficient of variation (CV) of inter-arrival times, and c_b is the CV of batch service time.

The first term, (k-1)/(2λ), is the average wait-to-batch time experienced by each single job. The second term is the queueing time from Kingman's G/G/1 approximation. Because each batch contains k jobs, and the job inter-arrival times are independent and

identically distributed, the squared CV of batch inter-arrival times is k times smaller than the squared CV of job inter-arrival times. The third term is the average batch service time.

To examine the effectiveness of this approximate model, we first compare the results of Eq. (3.1) with an $M/M^k/1$ queue, since an $M/M^k/1$ can be solved exactly (Gross and Harris 1998). In an $M/M^k/1$ queue, job arrivals occur as a Poisson process, and service times are exponentially distributed with a fixed parallel batch size k. The machine will process a batch if the batch size is exactly k. If the number of jobs in queue is less than k, they wait until k have accumulated. Batches are served on first-come-first-serve (FCFS) basis, and there is no limit on the queue length.

Under the assumption of an $M/M^k/1$ queue, Eq. (3.1) simplifies to

$$CT \cong \frac{k-1}{2\lambda} + (\frac{1/k+1}{2})BQT(M/M/1) + \frac{1}{\mu}, \qquad (3.2)$$

where

$$BQT(M/M/1) = (\frac{\rho}{1-\rho})\frac{1}{\mu}.$$

The queueing time in Eq. (3.2) is calculated based on Kingman's approximation, which is composed of a variability term, (1/*k*+1)/2, and an M/M/1 queueing time. We call the variability term a *G/G/1 transformer*, since it transforms an M/M/1 queueing time to a G/G/1 approximate queueing time. We call the M/M/1 queueing time a base queueing time (BQT). Thus, if the batch size is three, Eq. (3.2) can be illustrated by Figure 3.2.

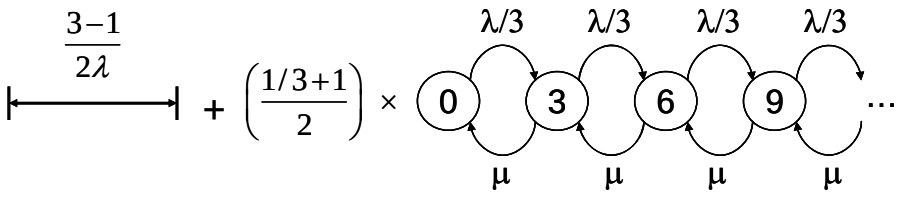

Figure 3.2 Graphical illustration of Eq. (3.2)

On the other hand, an $M/M^k/1$ queue is analyzed using a continuous time Markov chain model, which has a state transition rate diagram. When the batch size is three, the diagram is depicted in Figure 3.3.

The state diagram of Figure 3.3 is approximated by a flow equivalent birth and death process in Figure 3.2. The reasons for the differences between Figure 3.2 and Figure 3.3 are apparent: one is only an approximation, but the other one is the exact analysis. Investigating their differences may bring us valuable insight for further improvement on the current approximate model.

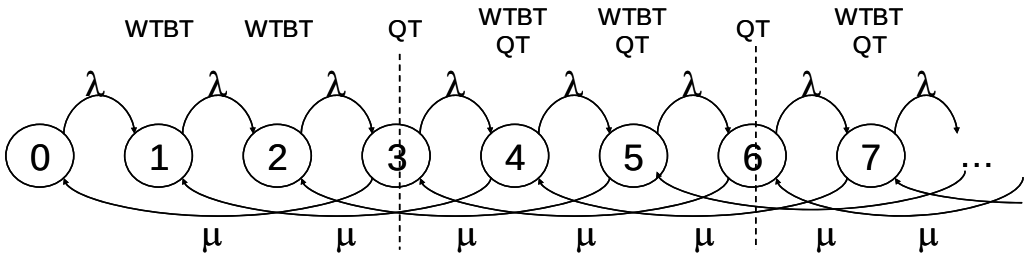

Figure 3.3 The state transition rate diagram of an $M/M^k/1$ queue

In Figure 3.3, since the batch size is three, the durations of state 1 and 2 are clearly the wait-to-batch times. The durations in state 3 and 6 are queueing times, since a complete batch is formed at those states which are the multiple of 3. Although the classification is clear at the above states, it is not so clear for the rest. For example, in states 4, 5 and 7, there are both wait-to-batch time and queueing time, since a complete batch is not formed yet, but there are indeed some formed batches in queue.

Wait-to-batch times and queueing times are not independent, at least not in states 4, 5, and 7! This observation suggests rethinking the suitability of Eq. (3.1), which ignores the overlap between wait-to-batch time and queueing time.

Comparing Figure 3.2 and Figure 3.3, the transition from state 4 to state 1 in Figure 3.3 has been ignored in Figure 3.2 (similar situation for state 5 to state 2). The

state changes can only occur at states 3, 6 and 9 in Figure 3.2. This implies the queueing time in Eq. (3.1) has been *overestimated*, since it takes longer than it should be to return to a lower state. Another source of errors comes from Kingman's approximation. As pointed by Shanthikumar and Buzacott (1980), Kingman's approximation overestimates the true value when the service time coefficient of variation is smaller than 1. However, this error becomes small when utilization is high. When service time variability is smaller than 1, both approximation errors overestimate the exact cycle time of the system. This tendency also can be observed in the simulation results of Fowler, et al. (2002), where they studied the multiproduct G/G/c model with batch processing.

To gain better understanding of the structured errors caused by Eq. (3.1), we have compared the approximate results from Eq. (3.2) and the exact results from an $M/M^k/1$ queue by a numerical example. Based on Gross and Harris (1998), the procedure to analyze an $M/M^k/1$ model is as follows:

(1) Solve for x in the characteristic equation (where $0 < x < 1$):

$$\mu x^{k+1} - (\lambda + \mu)x + \lambda = 0. \tag{3.3}$$

(2) Calculate the limiting probabilities p_0 and p_n,

$$p_0 = \frac{(1-x)}{k},$$

$$p_n = \begin{cases} \dfrac{p_0(1-x^{n+1})}{1-x} & (1 \le n < k), \\ \dfrac{p_0 \lambda x^{n-k}}{\mu} & (n \ge k). \end{cases}$$

(3) Calculate WIP, cycle time (CT), wait to batch time (WTBT), and queueing time (QT),

$$WIP = \sum_{n=1}^{\infty} nP_n,$$

$$CT = WIP / \lambda, \tag{3.4}$$

$$WTBT = \frac{k-1}{2\lambda},$$

$$QT = CT - WTBT - ST.$$

The definitions of parameters are the same as the parameters in Eq. (3.1). Rather than calculate p_0, and p_n, we may also get cycle time directly as follows,

$$CT = \frac{1}{\lambda k}\left(\frac{k(k-1)}{2} - \frac{x^2(1-x^{k-1})}{(1-x)^2} + \frac{(k-1)x^{k+1}}{1-x} + \frac{\lambda k}{\mu} + \frac{\lambda x}{\mu(1-x)}\right). \quad (3.4a)$$

One disadvantage of the above procedure is that it can only be solved numerically instead of explicitly, since we need solve Eq. (3.3) first. An alternative is to approximate x by a two term Taylor series expansion as follows,

$$x \cong 1 - \frac{2}{k+1}(1-\rho) - \frac{4}{3}\frac{(k-1)}{(k+1)^2}(1-\rho)^2. \quad (3.3a)$$

Eq. (3.3a) can greatly reduce the calculation effort, and gives accurate results when utilization is high. In the cases which we have examined, the errors of average cycle time are less than 3% as long as utilization is higher than 30%. The derivation of Eq. (3.3a) is given in Appendix B.

In the example, we assume the batch size (k) is 10, and μ is 300 min. Both service times and inter-arrival times are exponentially distributed. The results are summarized in Table 3.1. In Table 3.1, HQT and HCT are the queueing time and cycle time calculated based on Eq. (3.1). We first find that the error percentages of HCT, (HCT-CT)/CT, are all positive, which is consistent with our previous observation that Eq. (3.1) tends to overestimate the cycle times. Furthermore, the errors are smaller when the utilization becomes high or low. This means Eq. (3.1) may give us good approximations when the utilization is very high or very low. What are the reasons behind this regular pattern? Understanding this pattern of errors may provide insight for a better approximate model.

Table 3.1 Comparison between two models when k = 10 (Unit: min)

Utiliza-tion	Arrival Interval	Hopp and Spearman WTBT	HQT	HCT	$M/M^k/1$ WTBT	QT	CT	HCT Error %
10%	300.0	1350.0	18.3	1668.3	1350.0	0.3	1650.3	1.09%
20%	150.0	675.0	41.3	1016.3	675.0	5.6	980.6	3.63%
30%	100.0	450.0	70.7	820.7	450.0	21.4	771.4	6.39%
40%	75.0	337.5	110.0	747.5	337.5	50.5	688.0	8.66%
50%	60.0	270.0	165.0	735.0	270.0	97.6	667.6	10.10%
60%	50.0	225.0	247.5	772.5	225.0	174.8	699.8	10.39%
70%	42.9	192.9	385.0	877.9	192.9	306.3	799.1	9.85%
80%	37.5	168.8	660.0	1128.8	168.8	577.0	1045.8	7.93%
90%	33.3	150.0	1485.0	1935.0	150.0	1418.5	1868.5	3.56%
95%	31.6	142.1	3135.0	3577.1	142.1	3049.3	3491.4	2.45%

The errors in Eq. (3.1) mainly come from two sources. Kingman's heavy traffic approximation; and the missing transitions between (4, 1) and (5, 2) in Figure 3.3, etc., where (a, b) means the transition from state a to state b. Those missing transitions are represented as dashed lines in Figure 3.4. The reader may refer to Eq. (3.2) and Figure 3.2 for a better understanding of the two sources of errors.

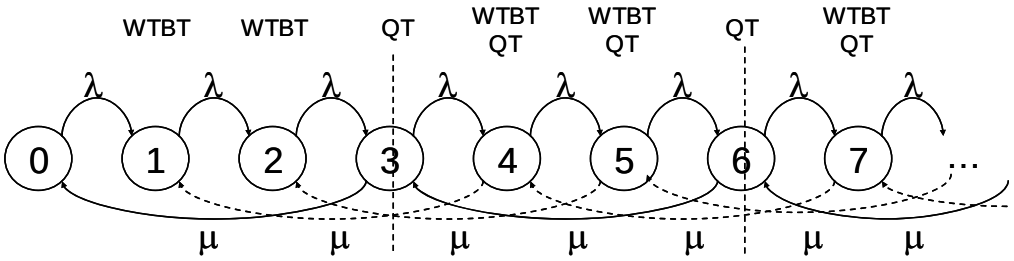

Figure 3.4 The state transition rate diagram of an $M/M^k/1$ queue

If a job arrives when the batch processing machine is idle, it will cause no errors in Eq. (3.1), since the machine is in states 0, 1 or 2, (not in 4, 5 and 7, etc.). This implies lower utilization would lead to less error, since machine has longer idle time at lower utilization. On the other hand, Kingman's approximation exhibits larger errors in queueing time at lower utilization. However, the error percentages (i.e. queueing time

errors / cycle time) from Kingman's approximation are relatively small compared with WTBT and service times, since queueing time itself is short. Therefore, these two sources of errors both tend to be small at low utilization.

If a job arrives when the machine is busy, the probability that it finds the machine in state 3t or 3t+1 is substantially greater than the probability it finds the machine in state 3t+2, where t is a natural number. If it arrives when the machine is in state 3t+2, it causes no errors, since all the transitions in 3t+3 are considered in Eq. (3.2). However, if it arrives when the machine is in state 3t or 3t+1, the state will become 3t+1 or 3t+2, respectively, which may cause errors, since compared with Figure 3.2, some of the transitions (presented by the dashed lines) are missing in those states. States 3t or 3t+1 are therefore called *incomplete states*.

Since the probability for a state to increase from t to t+1 is $\lambda/(\lambda+\mu)$ for t equal or greater than 3, the state will have higher probability to increase instead of decreasing when λ is larger. This transition will cause no errors, since it has been considered in Figure 3.2 (by considering the effect of λ's in Fig. 3.4 into the $\lambda/3$ in Fig. 3.2). However, on the other hand, if a job arrives at an incomplete state and its state decreases (with probability $\mu/(\lambda+\mu)$), it will cause errors, since this transition is missing in Figure 3.2. This observation suggests that higher utilization leads to smaller errors. At the same time, Kingman's approximation has smaller errors for higher utilizations. Therefore, these two sources of errors both tend to be small at high utilization.

From the above analysis, we know there are two opposite forces which affect the error percentages. To get smaller errors, one force prefers higher utilization, and the other prefers lower utilization. This explains why the error percentages become smaller at the both ends, high and low utilizations in Table 3.1.

When parallel batch size becomes larger, an incoming job may see an idle machine with higher probability, especially at lower utilization, which leads to smaller errors. Likewise, the errors from Kingman's approximation also become smaller due to

the relatively longer WTBT. On the other hand, an incoming job may also drop into the incomplete states with higher probability, which leads to larger errors. However, when utilization is high, the incoming state tends to increase (towards right), which would cause no errors, even if the job arrives at an incomplete state. At the same time, Kingman's approximation also gives smaller errors at high utilization.

Table 3.2 Comparison between two models when k = 5 (Unit: min)

Utiliza-tion	Arrival Interval	Hopp and Spearman			$M/M^k/1$				HCT
		WTBT	HQT	HCT	WTBT	QT	CT	ST	Error %
10%	600.0	1200.0	20.0	1520.0	1200.0	1.3	1501.3	300	1.25%
20%	300.0	600.0	45.0	945.0	600.0	10.6	910.6	300	3.78%
30%	200.0	400.0	77.1	777.1	400.0	31.2	731.2	300	6.28%
40%	150.0	300.0	120.0	720.0	300.0	65.5	665.5	300	8.19%
50%	120.0	240.0	180.0	720.0	240.0	118.9	658.9	300	9.28%
60%	100.0	200.0	270.0	770.0	200.0	203.5	703.5	300	9.45%
70%	85.7	171.4	420.0	891.4	171.4	349.4	820.8	300	8.61%
80%	75.0	150.0	720.0	1170.0	150.0	645.5	1095.5	300	6.80%
90%	66.7	133.3	1620.0	2053.3	133.3	1543.5	1976.8	300	3.87%
95%	63.2	126.3	3420.0	3846.3	126.3	3332.9	3759.2	300	2.32%

Therefore, when batch size increases, the errors at the middle utilization range will increase due to the increase of the incomplete states. The impact on low and high utilization will not be so significant. This phenomenon can be seen in Table 3.2, where the batch size is 5. The errors in the mid-range loading regime are smaller than the errors in Table 3.1 (reducing from 10.39% to 9.45% at 60% utilization) as expected.

3.3 The New Approximate Model

The previous G/G/1 based approximate model is convenient. However, systematic errors caused by overlap between waiting time and batching time exist in the model even for the $M/M^k/1$ case, where exact solutions are available. A new approximate model is proposed based on this new piece of information.

When arrival process is Poisson and service time is exponential, instead of getting the base queueing times from an M/M/1 model, we get the base queueing times (*BQT*) from the M/M^k/1 model by using Eq. (3.2) and Eq. (3.4) as follows,

$$CT(M/M^k/1) = \frac{k-1}{2\lambda} + (\frac{1/k+1}{2})BQT + \frac{1}{\mu}. \quad (3.5)$$

In this equation, we treat BQT as the unknown variable. CT(M/M^k/1) can be obtained from Eq. (3.4a), where *x* can be determined by solving either Eq. (3.3) or (3.3a). Therefore,

$$BQT = \frac{2}{1/k+1}\left(CT(M/M^k/1) - \frac{1}{\mu} - \frac{k-1}{2\lambda} \right). \quad (3.6)$$

Queueing times (QT) and cycle times (CT) can be obtained as follows,

$$QT \cong (\frac{c_a^2/k + c_b^2}{2})BQT, \quad (3.7)$$

$$CT \cong \frac{k-1}{2\lambda} + QT + \frac{1}{\mu}. \quad (3.8)$$

The definitions of parameters are the same as the parameters defined in Eq. (3.1). In Eq. (3.7), queueing times are the product of the base queueing times and the G/G/1 transformer.

Although the new approximation does not completely avoid the dependence between wait-to-batch time and queueing time, the error (caused by the dependence) can be, at least, kept to zero in the M/M^k/1 case. Thus, one can expect that the errors in the G/G^k/1 cases also can be reduced. The performance of this approximation will be tested by simulations in the next section.

3.4 Simulation Experiments

The improvement of the new approximate models is verified in three cases. In all three cases, the mean batch service times are 300 min, and the parallel batch size is 10.

To demonstrate the true improvement from the new model itself, and avoid errors from other factors, the $M/M^k/1$ cycle times are calculated based on the results from the standard procedure (i.e. Eq. (3.3)) instead of the Taylor series expansion (i.e. Eq. (3.3a)). However, when using Taylor series expansion, except for very low utilizations, only small errors are added into the model.

Table 3.3 Simulation results for Poisson arrival and constant service times (Case 1)

Utiliza-	Arrival	Simulation					Hopp and Spearman			New Approximation		
tion	Interval	SWTBT	90% CI	SQT	90% CI	SCT	WTBT	HQT	HCT	WTBT	QT	CT
10%	300.0	1350.2	0.04%	0.0	166.00%	1650.2	1350.0	1.7	1651.7	1350.0	0.0	1650.0
20%	150.0	674.8	0.05%	0.0	22.81%	974.8	675.0	3.8	978.8	675.0	0.5	975.5
30%	100.0	450.1	0.05%	0.0	4.86%	750.1	450.0	6.4	756.4	450.0	1.9	751.9
40%	75.0	337.4	0.05%	0.3	1.81%	637.8	337.5	10.0	647.5	337.5	4.6	642.1
50%	60.0	270.0	0.04%	1.4	0.95%	571.5	270.0	15.0	585.0	270.0	8.9	578.9
60%	50.0	224.9	0.04%	4.6	0.57%	529.5	225.0	22.5	547.5	225.0	15.9	540.9
70%	42.9	192.9	0.04%	12.5	0.43%	505.4	192.9	35.0	527.9	192.9	27.8	520.7
80%	37.5	168.8	0.04%	32.9	0.49%	501.7	168.8	60.0	528.8	168.8	52.5	521.2
90%	33.3	150.1	0.05%	103.3	0.74%	553.3	150.0	135.0	585.0	150.0	129.0	579.0
95%	31.6	142.1	0.04%	250.8	1.28%	692.9	142.1	285.0	727.1	142.1	277.2	719.3

The first case we examined has Poisson arrival and constant service times. The results are shown in Table 3.3. As in Table 3.1, HQT and HCT are the queueing times and cycle times calculated based on Eq. (3.1). SWTBT, SQT and SCT are the wait-to-batch times, queueing times and cycle times from simulation. At each specific utilization level, each WTBT, SQT and SCT is the mean of 100 replications. In each replication, we collected data for 200,000 jobs after a 50 year warm-up period, (thus, the number of jobs discarded in the warm-up can be approximated as "50 years * 365 days * 1440 min / arrival-interval"). The above parameters are chosen to reduce the confidence interval, but within a tolerable simulation run time. The half-width of 90% confidence intervals (CI) are listed right after the corresponding simulation values. Service times are omitted in the table, since they are all about 300 min, and the 90% CI are all smaller than 0.1%. The units of wait-to-batch times, queueing times, and cycle times are minutes. Since the

service times are constant (c_b is 0), the queueing times tend to be small compared with the mean service times, even at high utilization.

Table 3.4 Simulation results for Poisson arrival and Erlang-2 service times (Case 2)

Utiliza-	Arrival	Simulation					Hopp and Spearman			New Approximation		
tion	Interval	SWTBT	90% CI	SQT	90% CI	SCT	WTBT	HQT	HCT	WTBT	QT	CT
10%	300.0	1350.4	0.05%	0.0	15.18%	1650.3	1350.0	10.0	1660.0	1350.0	0.2	1650.2
20%	150.0	675.1	0.04%	1.1	2.13%	976.0	675.0	22.5	997.5	675.0	3.1	978.1
30%	100.0	450.1	0.04%	6.1	1.00%	756.4	450.0	38.6	788.6	450.0	11.7	761.7
40%	75.0	337.4	0.04%	17.9	0.73%	655.6	337.5	60.0	697.5	337.5	27.5	665.0
50%	60.0	269.9	0.04%	39.7	0.73%	609.5	270.0	90.0	660.0	270.0	53.2	623.2
60%	50.0	225.1	0.04%	77.8	0.56%	602.9	225.0	135.0	660.0	225.0	95.3	620.3
70%	42.9	192.9	0.04%	147.0	0.69%	639.8	192.9	210.0	702.9	192.9	167.1	659.9
80%	37.5	168.7	0.04%	290.6	0.72%	759.2	168.8	360.0	828.8	168.8	314.7	783.5
90%	33.3	150.0	0.04%	733.5	1.24%	1183.5	150.0	810.0	1260.0	150.0	773.7	1223.7
95%	31.6	142.1	0.04%	1626.6	2.02%	2068.6	142.1	1710.0	2152.1	142.1	1663.3	2105.4

In the second case, the arrival process is still Poisson, but service times follow an Erlang-2 distribution. The squared coefficient of variation (SCV) of service times is 0.5. Comparing Case 2 with Case 1, since the service times are changed from constant to Erlang-2 distribution, the queueing times become considerably longer compared with the mean service times.

Table 3.5 Simulation results for Erlang-10 arrivals and Erlang-2 service times (Case 3)

Utiliza-	Arrival	Simulation					Hopp and Spearman			New Approximation		
tion	Interval	WTBT	90% CI	SQT	90% CI	SCT	WTBT	HQT	HCT	WTBT	QT	CT
10%	300.0	1343.3	0.83%	0.0	166.00%	1643.5	1350.0	8.5	1658.5	1350.0	0.1	1650.1
20%	150.0	675.1	0.01%	0.1	6.53%	975.3	675.0	19.1	994.1	675.0	2.6	977.6
30%	100.0	448.9	0.41%	2.2	6.30%	751.2	450.0	32.8	782.8	450.0	9.9	759.9
40%	75.0	337.5	0.01%	9.2	0.98%	646.7	337.5	51.0	688.5	337.5	23.4	660.9
50%	60.0	270.0	0.01%	25.3	0.70%	595.5	270.0	76.5	646.5	270.0	45.2	615.2
60%	50.0	225.0	0.01%	55.3	0.71%	580.3	225.0	114.8	639.8	225.0	81.0	606.0
70%	42.9	192.8	0.01%	111.9	0.70%	604.6	192.9	178.5	671.4	192.9	142.0	634.9
80%	37.5	168.6	0.18%	237.8	3.38%	706.3	168.8	306.0	774.8	168.8	267.5	736.3
90%	33.3	150.0	0.01%	598.8	1.41%	1048.5	150.0	688.5	1138.5	150.0	657.7	1107.7
95%	31.6	142.1	0.01%	1370.6	2.51%	1812.7	142.1	1453.5	1895.6	142.1	1413.8	1855.9

In the third case, the service times still follow an Erlang-2 distribution, but the arrival intervals follow an Erlang-10 distribution. The SCV of arrival intervals is 0.1.

The errors of estimated queueing times from the old and new models are shown in Table 3.6. HQT error is "HQT/SQT – 1", and QT error is "QT/SQT – 1". Improvement is "HQT error/QT error – 1", which gives the improvement of the new model. In all three cases, both old and new models give large errors (HQT error and QT error) at low utilization and relatively small errors at high utilization, since Kingman's formula is a heavy traffic approximation.

Table 3.6 Errors of the three models by using Eq. (3.3)

		Case 1: $c_a^2 = 1$, $c_b^2 = 0$			Case 2: $c_a^2 = 1$, $c_b^2 = 0.5$			Case 3: $c_a^2 = 0.1$, $c_b^2 = 0.5$		
Utilization	Arrival Interval	HQT Error	QT Error	Improvement	HQT Error	QT Error	Improvement	HQT Error	QT Error	Improvement
10%	300.0	8124942.8%	127787.1%	98.4%	49125.9%	674.8%	98.6%	400452.9%	6204.7%	98.5%
20%	150.0	344446.7%	46975.6%	86.4%	2035.4%	191.8%	90.6%	15569.3%	2040.9%	86.9%
30%	100.0	16519.6%	4928.6%	70.2%	534.2%	91.9%	82.8%	1413.6%	358.0%	74.7%
40%	75.0	3050.1%	1344.8%	55.9%	234.4%	53.4%	77.2%	454.8%	154.5%	66.0%
50%	60.0	961.5%	527.7%	45.1%	126.6%	34.0%	73.1%	202.4%	78.8%	61.1%
60%	50.0	388.0%	244.7%	36.9%	73.5%	22.5%	69.3%	107.7%	46.7%	56.6%
70%	42.9	179.8%	122.6%	31.8%	42.8%	13.6%	68.2%	59.5%	26.9%	54.8%
80%	37.5	82.2%	59.3%	27.9%	23.9%	8.3%	65.2%	28.7%	12.5%	56.4%
90%	33.3	30.7%	24.8%	19.1%	10.4%	5.5%	47.4%	15.0%	9.8%	34.4%
95%	31.6	13.7%	10.5%	22.8%	5.1%	2.3%	56.0%	6.0%	3.1%	47.9%

The improvement percentage of the new model decreases with increasing utilization. In Case 1, the improvement from the new models decreases from 98% (at 10% utilization) to 23% (at 95% utilization). Furthermore, among these three cases, the improvement increases when either c_a^2 or c_b^2 is close to one. This observation is consistent with our assumptions, since we know our approximate model will give exact solutions when the service time is exponential and the arrival process is Poisson. On the other hand, this new model may not perform very well when c_a^2 and c_b^2 are much larger than one.

However, in practical manufacturing systems, to maintain competitiveness,

service time SCV is desired to be small. Therefore, c_b^2 is chosen to be 0 in Case 1 and 0.5 in Case 2 and 3. Among the three cases, due to the Palm-Khintchine theorem, Case 2 may be representative to the situations when the machine is fed by multiple upstream workstations, and each workstation is composed of multiple machines. If the machine is only fed by one or two upstream machines, the c_a^2 can be small. Case 3 may be representative in this situation. As we have seen in Table 3.6, in both cases, the original errors can be around 10% and the improvement can be around 50% at high utilization.

If we use Eq. (3.3a), the Taylor series expansion approximation, the errors will be larger at low utilization, but almost the same at high utilization (see Table 3.7). In the examined cases, because the value of x is overestimated at the low utilization by the Taylor series expansion, the estimated queueing time indeed becomes negative when utilization is less than 20%. However, it causes no significant impact to the overall system, since the true queueing time in this situation is less than two minutes compared with the WTBT of 1350 minutes.

Table 3.7 Errors of the three models by the Taylor series expansion

		Case 1: $c_a^2 = 1$, $c_b^2 = 0$			Case 2: $c_a^2 = 1$, $c_b^2 = 0.5$			Case 3: $c_a^2 = 0.1$, $c_b^2 = 0.5$		
Utiliza-tion	Arrival Interval	HQT Error	QT Error	Improve-ment	HQT Error	QT Error	Improve-ment	HQT Error	QT Error	Improve-ment
10%	300.0	8124942.8%	--	--	49125.9%	--	--	400452.9%	--	--
20%	150.0	344446.7%	--	--	2035.4%	--	--	15569.3%	--	--
30%	100.0	16519.6%	4707.9%	71.5%	534.2%	83.5%	84.4%	1413.6%	337.9%	76.1%
40%	75.0	3050.1%	1704.4%	44.1%	234.4%	91.5%	60.9%	454.8%	217.8%	52.1%
50%	60.0	961.5%	642.2%	33.2%	126.6%	58.5%	53.8%	202.4%	111.5%	44.9%
60%	50.0	388.0%	278.9%	28.1%	73.5%	34.7%	52.8%	107.7%	61.2%	43.1%
70%	42.9	179.8%	134.1%	25.4%	42.8%	19.5%	54.5%	59.5%	33.4%	43.8%
80%	37.5	82.2%	62.5%	24.0%	23.9%	10.5%	56.1%	28.7%	14.7%	48.6%
90%	33.3	30.7%	23.6%	23.1%	10.4%	4.4%	57.5%	15.0%	8.7%	41.6%
95%	31.6	13.7%	10.6%	22.7%	5.1%	2.3%	55.8%	6.0%	3.2%	47.7%

3.5 Conclusion

By detailed examination of the existing G/G/1 based approximate models for parallel batch processing, the information lost during the decomposition process has been identified. By partially recovering the lost information, a new approximate model has been proposed. The new model shows notable improvement over the previous approaches. However, it only considers the case of single servers. The approximate model considering a workstation with multiple servers is left for future research.

Although the main focus of this chapter is analyzing parallel batch processes, some types of interruptions which we have discussed in Chapter 2 can be readily incorporated, such as run-based preemptive interruptions and run-based non-preemptive product-induced interruptions. Since those two events only occur during processing, we can consider their impacts on the system by adjusting the service time distribution directly.

The scenario described in this chapter indeed gives us a good example to explain the potential risk of applying decomposition. Decomposition is a powerful and convenient technique, especially when we want to analyze large complex systems in a practical environment. However, in complex manufacturing systems, the independence assumption inherent in decomposition often is not satisfied. As a result, the approximation errors from decomposition can be significant if the potential for information loss is not recognized and dealt with, as illustrated in this chapter.

If the dependence among components is too strong, maybe it is appropriate to rethink the applicability of decomposition, since to recover the lost information during the decomposition may be more involved than to deal with the system as a whole. Furthermore, sometimes, the lost information during the decomposition may be the key to understanding the behavior of the whole system. More examples will be given in part II, where we attempt to describe the behavior of manufacturing systems.

PART II

BEHAVIOR OF MANUFACTURING SYSTEMS

CHAPTER 4

ASSUMPTIONS, PREVIOUS WORK AND MOTIVATION

Perfection of means and confusion of goals seem to characterize our age.

~ Albert Einstein

4.1 Introduction

In part I, the behavior of a single machine has been examined. In part II, we will study the behavior of manufacturing systems, which are composed of a series of workstations (with or without feedback), and each workstation is composed of a single or multiple machines. This type of manufacturing system can be modeled as a queueing network, where each node represents a workstation.

The behavior of two single-server queues in series will be discussed in Chapter 5. Chapter 6 will introduce the behavior of many-server queues in series. Characterizing the performance of general manufacturing systems will be introduced in Chapter 7. Before we start our discussion on the above topics, we first explain our motivation by a case study, give a literature review in Section 4.2, and discuss the major assumptions in Section 4.3.

In order to understand the behavior of manufacturing systems, we will first study the behavior of two single server queues in series in Chapter 5. As stated in Tembe and Wolff (1974), "A system of tandem queues can be described as a service facility with two or more stations in series." Therefore, we call this system a simple tandem queue. Since a simple tandem queue is a combination of two single servers, its behavior is more complex than what we have seen in Part I. The objective is to understand the dependence between the two servers. Compared to a genuine manufacturing system, a simple tandem queue is much simpler, and it is one of the simplest systems possessing dependence

among workstations. Understanding its behavior, especially the dependence between stations, can give us useful insight towards understanding the behavior of more realistic models of manufacturing systems.

A simple tandem queue consists of two single servers in series. If the two servers are independent of each other, the overall system performance would be the gross performance of each system as they see the initial arrival process directly. Therefore, independence is a very nice property when it holds. Although independence does not hold in general for manufacturing systems, it is interesting to ask under what condition it may hold. This condition was not identified until the extraordinary work of Jackson (1957) who considered networks with exponentially distributed service times and inter-arrival times. A queueing network which possesses this nice property is therefore called a Jackson network.

In the case of a simple tandem queue, independence holds if and only if the service time and inter-arrival time are exponentially distributed with FIFO dispatching scheme (Daley 1968). Studying the behavior of a series of more general workstations is in the scope of queueing networks. As we have seen in Part I, understanding the behavior of a single server system is already not an easy task. It is obvious that studying the behavior of queueing networks will be much more involved. In particular, the workstations are not independent in general.

4.1.1 A Case Study

Figure 4.1 shows two snapshots of partial WIP profiles from a semiconductor fab at two time epochs. The time between these two snapshots is about a month. The WIP level at each process step is represented by a rectangular bar. The process step names would be under those bars, but they have been removed to maintain confidentiality of the data except for a few that are crucial for our explanation.

In the first snapshot (the upper one), we find WIP is mainly accumulated at lithography (DT, AA, WB and DC PH), which is the designed bottleneck of the fab. One month later (the lower one), the WIP at DT PH drops from ~5,400 to less than 500. (Notice that the scales of these two snapshots are different.) The WIP bubble at DT photo is pushed to its downstream workstations.

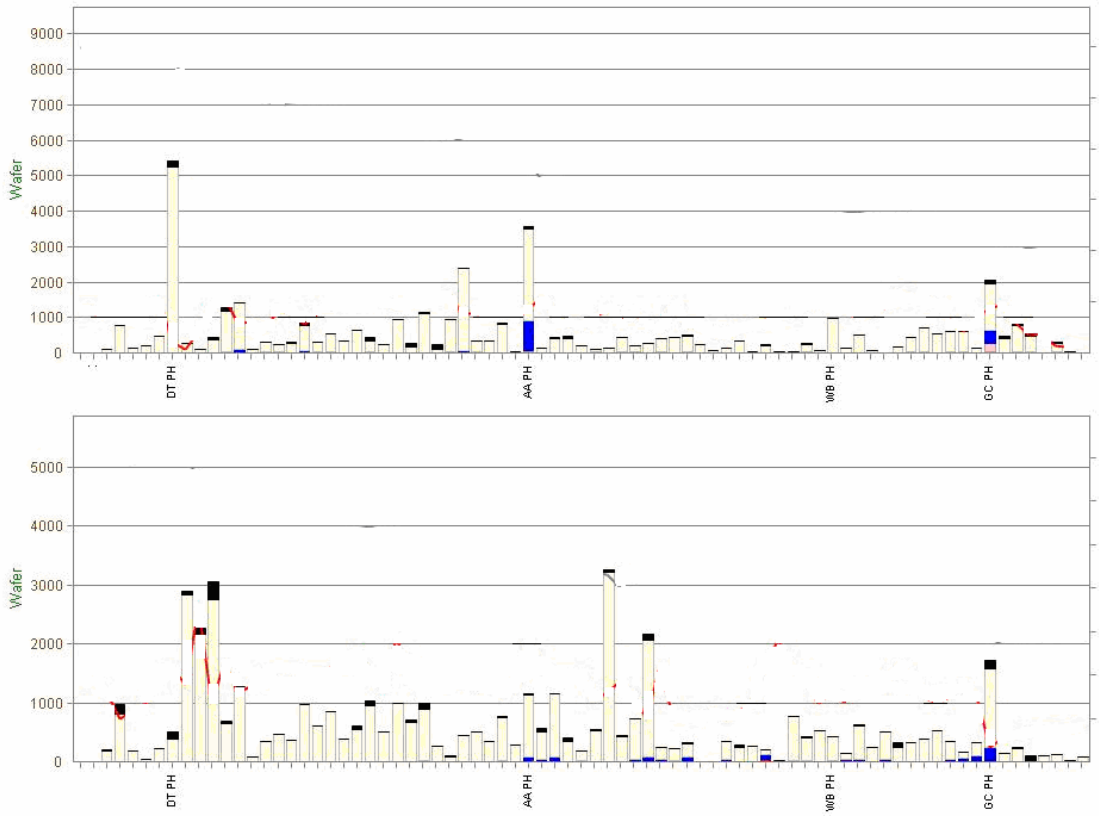

Figure 4.1 Two snapshots of WIP profiles from a semiconductor fab

From the scenario presented in Figure 4.1, the following interesting observations can be obtained:

1. The movement of WIP bubbles can be very slow. This can be observed by comparing the changes between two snapshots.
2. WIP distributes unevenly in a fab. One of the main reasons is because capacity is inversely related to process cost, and the cost of each machine is different. In

practice, we will find bottlenecks at the most expensive machines. Their utilization will be the highest, and WIP is also the highest in heavy traffic.

3. There is dependence among workstations. For example, workstations located before or after the bottleneck will have different WIP distributions. Furthermore, when WIP is blocked at one workstation, its downstream workstations may starve due to the resulted low WIP levels.
4. The WIP at DT PH can drop from ~5,400 to less than 500, while some others can increase from less than 300 to more than 3,000. The WIP buffer size is large, and does not limit the large WIP fluctuations at those workstations.

It should be noted that, due to the second observation, the capacity of two consecutive workstations usually will be different. It also is important to note the critical role of dependence in a fab. We could not understand the behavior of a fab without capturing dependence in our models.

4.2 Literature Review

Queueing networks can be classified as two kinds: open or closed queueing networks. In an open queueing network, customers can arrive or leave the system at any node, while in a closed queueing network, only a fixed number of customers continuously travel inside the queueing network.

The study of queueing networks can be traced back to Jackson (1957). He proves that in an open network, when the inter-arrival time and service time are exponentially distributed, the steady state solutions have a closed product form for the stationary multi-dimensional state probabilities, where the factors are the solutions of isolated exponential queueing stations. In other words, the system behaves as if each workstation acts independently in the steady state.

After the pioneering work of Jackson, research in queueing networks flourished. Research in this field mainly centered around the extension of Jackson's product form result, and can be classified into two main approaches: exact analysis and approximation methods.

4.2.1 Exact Analysis

Exact analysis explores closed-form solutions of queueing network models. Although, in general, we cannot expect exact closed-form solutions in practical queueing networks, they do exist in some special cases. Understanding those special cases may help us to gain insights of the underlying structure of more general queueing networks.

In 1957, Jackson derived exact product form solutions for an open Jackson network. After Jackson's seminal contribution, Gordon and Newell (1967) extended the results to closed queueing systems. They showed that the behavior of such closed systems is stochastically equivalent to open systems in which the number of customers is conditioned to be equal to the same amount of customers in the closed system. Baskett, Chandy, Muntz, and Palacios (1975) further extended Jackson's basic product form solutions to open, closed, and mixed queueing networks with different classes of customers for exponential service times under FCFS, or phase-type service times under processor-sharing, infinite servers systems and preemptive-resume LCFS. They assume that the arrival process is Poisson (or a superposition of m Poisson arrival streams). The above generalized networks are also called BCMP networks.

Reiser and Lavenberg (1980) demonstrated that mean value analysis (MVA) can be used to describe the system behavior of closed multiple-chain queueing networks which have product-form solutions by computing recursively without computing product terms and normalization constants.

In 1965, two important papers, by Friedman (1965) and Avi-Itzhak (1965), were published in two journals at the same time, which both investigated the behavior of a tandem queueing system with constant service times. Avi-Itzhak showed: (a) the time spent in the system by any customer is independent of the order of the stations and of the allowable sizes of the intermediate queues; (b) The queueing time of any customer equals the time the same customer would have been waiting in the queue of a single server system with constant service time, equaling the longest service time of the sequence. On the other hand, Friedman showed if customers arrive at the first stage and proceed though the stages in order of FCFS with infinite buffers: (a) For any sequence of customer arrival times, the time spent in the system by each customer is independent of the order of the stages; (b) Under certain conditions, a tandem queueing problem could be reduced to corresponding problems for a system of fewer stages, possibly a single stage. This procedure is called a reduction method.

In contrast with the independence property of Jackson networks, Friedman and Avi-Itzhak showed that a tandem queueing system with constant service times also possesses very nice properties: The total system queueing time is solely determined by the bottleneck workstation. In other words, the queueing time of any customer equals the time the same customer would have been waiting in the queue of a single workstation, which is the bottleneck workstation of the tandem queue system. Therefore, we can analyze a queueing system exactly without the independence assumption.

Thus, product form solutions are not the only way to get closed form solutions of queueing networks. While Jackson's finding lead to the powerful theory of product form networks, we point out in this thesis that Friedman's exact result can be used to incorporate network dependence. Later we will show how to rely on the properties of reduction methods to identify the underlying structure of general queueing networks.

Extending Friedman's results, Tembe and Wolff (1974) gave the optimal order of service in tandem queues. Although the optimal order only exists under some specific

conditions, such as non-overlapped service times, the results are exact once the conditions are satisfied.

In the area of exact analysis, although researchers made impressive progress from Jackson networks to BCMP networks and beyond (a state of the art analysis can be found in Serfozo (1999), and Chen & Yao (2001)), compared with the real situations in any practical manufacturing system involving intricate dependencies, the conditions are still far from general. Therefore, to understand the performance of practical manufacturing systems, researchers often resort to approximation methods.

4.2.2 Approximation Methods

Although there are many types of approximation methods for queueing networks, the main development of approximation methods can be classified into the following three categories: decomposition methods, diffusion approximations and mean value analysis.

Decomposition methods are used to approximate queueing network performance by breaking the network into subsystems which are analyzed in isolation. Chandy, Herzog and Woo (1975) presented an approximate iterative technique for the analysis of general queueing networks. The technique determines approximations of the queue length and waiting time distributions for each queue in the network. Although the authors pointed out that approximating a general network by an exponential network usually results in unacceptable errors, the algorithm is adequate for the configuration phase of computer and teleprocessing system design, and is limited only by the kinds of reduced networks that are analyzable.

In order to make the behavior of a network tractable, Kleinrock (1976) proposed the independence assumption which introduced the concept of exponential queueing networks. Kuehn (1979) proposed a decomposition method to approximate the

performance of a general open queueing network. The network is composed of N single server queueing workstations with arbitrary interconnections. The analysis is based on decomposition where the network is broken up into N G/G/1 subsystems. The subsystems are analyzed individually by assuming renewal arrival and departure processes. Shanthikumar and Buzacott (1981) proposed an approximate decomposition approach for an open queueing network model of dynamic job shops with general service times and FCFS or shortest processing service discipline, where each workstation in the job shop is still composed of a single server.

Decomposition methods for queueing networks with multiple servers were proposed by Whitt (1983). Whitt's Queueing Network Analyzer (QNA), classified as parametric-decomposition approximations, is an analytical tool based on mathematical formulae and simple approximations rather than involved numerical procedures. Because the fundamental structure of QNA is derived based on Kingman's equation (Heyman 1975), it can also be considered as a heavy traffic approximation. QNA analyzes open networks of multi-server queues with the FCFS discipline and no capacity constraints. The exogenous arrival interval and the service time can be generally distributed. QNA uses two parameters to characterize the arrival processes and service times; one to describe the rate and the other to describe the variability. The nodes are then analyzed as standard G/G/m queues partially characterized by the first two moments of the inter-arrival time and service time distribution. Originally, QNA was designed for the analysis of teletraffic networks. To extend its application to manufacturing systems, Segal and Whitt (1989) have modified QNA to incorporate machine breakdowns, batch services, changing lot sizes and product testing with associated repair and partial yields. Since QNA considers generally distributed service times and inter-arrival times by variability parameters, its capability to describe the behavior of general queueing networks is better than that of the standard Markovian models.

Diffusion approximation methods use diffusion processes to analyze the behavior of queueing networks in heavy traffic. Kobayashi (1974) introduced the usage of such processes to the application area of computer networks. A vector-valued diffusion process and its diffusion equation are introduced to approximate the queueing processes in a general queueing network. Equilibrium queue distributions and transient behaviors are discussed in two of his papers. Reiser and Kobayashi (1974) use the diffusion process approximation to develop realistic analytical models of computing systems by considering service time distributions of a general form. It is based on the assumption that queues are almost always nonempty and, therefore, the queue size fluctuations are normally distributed. They used a computationally simpler but less exact approach than the methods introduced before. In general, these methods perform well for networks in heavy traffic, and considerable better than the exponential server model.

Motivated by heavy traffic theory, Harrison and Nguyen (1990) proposed the QNET method to approximate queueing networks. They focus on the stationary distribution of these diffusion processes for steady-state analysis of the corresponding queueing systems. QNET uses multidimensional reflected Brownian motion (RBM), also called regulated Brownian motion, on the k-dimensional nonnegative orthant to approximate a k-workstation queueing network. Dai and Harrison (1992) developed the QNET algorithm to obtain the numerical results. However, the computational complexity of the QNET algorithm grows in the size of the network.

In order to overcome the issues from computational complexity and the heavy traffic assumption associated with QNET, Dai, Nguyen and Reiman (1994) developed the Sequential Bottleneck Decomposition (SBD) method to reduce the computational complexity by grouping workstations with similar utilizations, and limiting these sub-networks to a reasonable size. For the few cases they have examined, the results showed that SBD and QNET outperform QNA when there is intensive short-loop feedback (i.e. 50% feedback within 2 steps) in the networks. However, when feedback does not exist,

QNA performs better in four out of the six examined cases (2 nine-station and 4 ten-station cases).

A potential issue in applying heavy traffic approximations in queueing networks is its dependence on the central limit theorem. Because inter-arrival times are correlated in general queueing networks, the step size of random walks are correlated. Thus, the "identically and independently distributed" assumption of the central limit theorem is violated. Based on the simulation experiments in Chapter 5 and 6, the results from the Brownian approximation can be unreliable in this situation.

Mean value analysis (MVA) can be used to compute the performance measures of closed Jackson queueing networks without computing product terms and normalization constants (Reiser and Lavenberg 1980). The associated algorithm is constructed based on two assumptions: (1) an arriving job "sees" the system with itself removed in equilibrium; (2) Little's law can be applied to the queueing network. The first assumption is true only when service times are exponential. Therefore, each server works independently in steady state. The storage requirements for intermediate results in MVA increase rapidly for networks with multiple server workstations, customer classes and number of customers within each class. In order to reduce the computational complexity of MVA, some approximation methods have been developed. For networks with single server workstations, Schweitzer (1979) proposed an approximation which has been extensively tested by Bard (1979). This so called Schweitzer-Bard approximation improves the computational efficiency of the MVA algorithm for network with single server workstations. Suri, Sahu and Vernon (2007) extend Schweitzer-Bard's MVA approximation from single-server workstations to multiple-server workstations networks.

In summary, the MVA algorithms are designed by assuming each workstation behaves independently in steady state. Therefore, it is only exact for a product form network. If the servers are not independent in steady state, the results are only approximations. Diffusion approximations assume the validity of the central limit

theorem and focus on the behavior of queueing networks in heavy traffic. They start by approximating the behavior of single server, then extending the results to the network. The errors increase when utilization decreases or the correlations among inter-arrival times increase. On the other hand, decomposition methods are relatively simple to implement, but fail to capture the dependence among servers. As we will see in Chapter 5, both QNET and QNA perform poorly when the variance of service times is close to zero.

Although MVA and QNET approximate queueing network performance from the viewpoint of the whole system, both are derived based on strong assumptions and suffer computational complexity issues. On the other hand, QNA adopts the decomposition approach rather than the whole system view. Although decomposition also induces errors by neglecting dependence, its calculation is relatively simpler than the other two.

4.3 Major Assumptions and Justifications

We now provide useful background to justify the assumptions we are going to make later. To illustrate the need for this, we elaborate on a real world example.

Ovens or furnaces are one of the commonly seen machine types in real production lines. Although industrial ovens may achieve higher temperature and stability, their function is indeed similar to the ovens we use in our homes. When you want to roast a chicken in your oven, do you set up the service time to be constant or exponential? The answer is obvious, and we probably do not know how to set up an exponential service time on our oven. However, due to some unavoidable random fluctuations, the service times could be a little bit longer or shorter than the pre-determined recipe time. The real distribution could be similar to a triangular distribution with small deviations, where a triangular distribution is a continuous distribution defined on the interval [a, c] with mode b. However, even when the deviations are substantial, for a triangular distribution, triang(a, b, c), the largest squared coefficient of variation (SCV) of a symmetric positive

triangular is only 1/6 (i.e. when a = 0, c = 2b). If the distribution of service times is uniform, U(a, b), the largest SCV is only 1/3 (i.e. a = 0). In real situations, the SCV could be even smaller, since their distribution will not start from 0. Then, why is the exponential distribution often assumed for service times (with SCV = 1)?

One of the key factors which increases the variability of service times is interruptions. We have discussed this in Part I. Since interruptions cause negative impact on productivity, all fabs try to eliminate interruptions as much as possible. Although we could decrease the impact of interruptions through continuous improvement programs, we cannot totally eliminate them.

There are two main types of interruptions: preemptive and non-preemptive. The preemptive interruptions do add some randomness to the service times. Furthermore, there are two types of preemptive interruptions: run-based and time-based. Their impacts on downstream machines can be different. However, to simplify our models, we will focus on run-based preemptive interruption first. Therefore, the first major assumption is that *all interruptions are run-based preemptive.* This assumption will be used in Chapter 5 and 6, but will be relaxed in Chapter 7.

However, as we have seen in SEMI E10, in a semiconductor fab, the majority of interruptions are non-preemptive in nature. For the non-preemptive interruptions, we often have some levels of control on their occurrences. That means we may have the choice to postpone them to minimize the impacts on production lines. For example, for a run-based non-preemptive interruption, like a setup, we usually try to reduce its frequency, and do a changeover for a specific product only when it has considerable number of waiting jobs. For a time-based non-preemptive interruption, like a preventive maintenance (PM), we may postpone the execution of a PM until a time when the machine is less busy. In the auto industry, some factories in the US even have a two hour break after 10 hours of work. During those two hours, the whole production line is stopped for maintenance or changeover if necessary. The randomness caused by non-

preemptive interruptions is greatly reduced in this case. Then, why do we assume exponential service times, which have a squared coefficient of variation equal to one?

Another important factor which may cause randomness of service times is the product mix. Among all factories, semiconductor foundry fabs may have the most complex product mixes and process flows. In some old fabs of the first tier foundry companies, there can be more than 200 products which are run in the production lines concurrently. Many products have the same or similar process flows, except that they may use different masks at the photolithography. A process flow can be viewed as an aggregation of a long series of process steps, where a specific recipe is assigned to each step. The process steps of the same or different process flows may use the same recipes. Therefore, while the products can be more than 200 and the process flow can be long, when they are down to the recipe level, each workstation is usually responsible for only a few production recipes in a certain period of time.

Since a workstation is usually composed of multiple servers/machines, the products with the same or different (but similar) recipes are often assigned to dedicated servers, if they do not need extra setup among those recipes. Obviously, the recipe times are the same for products with the same recipe. However, even if their recipes are different, the recipe times still can be the same or similar, if they do not need extra setup among those recipes. Therefore, if we look at one single server, its service times can be very regular in a certain period of time. Then, why do we assume exponential service times?

In 1996 IEEE/SEMI ASMC, F. G. Boebel reported that the SCV of logic products in a joint SIEMES/IBM plant is 0.5, while the SCV of memory products is 0.4. Based on the above reasons, it is reasonable to begin our study of a manufacturing system with assuming *the SCV of service times is smaller than 1 (and perhaps much smaller)*. This is indeed our second main assumption in Part II. We will only briefly discuss the behavior of manufacturing systems which have service time SCV greater than 1.

The third assumption we impose concerns buffer sizes. Many previous researchers focus on the behavior of queueing networks with blocking. A special issue of the Annals of Operations Research in 1998, edited by Y. Dallery and D. D. Kouvatsos, has been dedicated on this topic. However, in our study, we assume that *there is no buffer limit between any two consecutive workstations* in a fab due to the following three reasons: First, in high-tech manufacturing facilities, WIP storage cost is considerable less than the cost of machines. Since the machine can be very costly, it may not be wise to hurt machine productivity by saving money on WIP storage. Secondly, if the assigned stocker (or WIP rack) is full, a job can be stored in other neighbor stockers, which are not full. Thirdly, instead of exploding the manufacturing systems by WIP, production planners will control the total amount of WIP at a pre-determined level by the job release policy.

Indeed, the buffer limit in practice is a soft constraint instead of a hard one. In practice, the upstream workstations will avoid flooding one particular downstream machine by processing jobs for other downstream machines with fewer waiting jobs. However, if all the downstream machines are equally full, it can arbitrarily send jobs to one of them by allocating the jobs to their neighbor stockers. Therefore, the buffer size is seldom a constraint in practice.

The fourth assumption is: *The dispatching rule is first-come-first-serve* (*FCFS*). Although it is not true in general, we assume it in Chapter 5 and 6 to simplify the model derivations. However, it will be relaxed in Chapter 7, where the new model will be tested in manufacturing systems with dispatching rules other than FCFS.

In practice, the downstream WIP levels (or queueing times) can be regulated by dispatching rules. Therefore, the fluctuations of WIP levels would be smaller than the situations without control. Due to the fourth assumption, we might expect the real queueing times to be shorter than the prediction from our models.

There are two kinds of feedback in semiconductor fabs: rework and reentry. Rework occurs when the quality of product at a workstation does not satisfy the specifications. The feedback to the workstation is often immediate; the process is repeated until it meets the specifications. Since rework is a total waste of capacity, the goal is to make the rework rate as low as possible. A new process usually will not be released for production if the rework rates are too high. The rework rate is usually a little bit higher at the beginning of a new process and becomes lower with experience.

On the other hand, reentry is inherent to semiconductor processes. In general, every product needs to visit lithography at least 20 ~ 30 times before its completion. Based on Leachman and Hodges (1996), the cycle times between two consecutive visits are usually 2 ~ 4 days. Within this duration, it usually contains 10 ~ 20 process steps. The dependence caused by reentry is therefore weak.

Feedback is in general not a dominant property in manufacturing systems. It is therefore reasonable to start by comparing our new model with other models without feedback. Therefore, the fifth assumption in chapters 5 and 6 is: *there is no feedback*. In chapter 7, we extend the results from Chapter 5 and 6 to allow short-loop feedback.

4.3.1 Decomposition of System Cycle Time

If all service times are constant in tandem queues, Avi-Itzhak (1965) shows "the queueing time of any customer equals the time the same customer would have been waiting in the queue of a single server system with constant service time, equaling the longest service time of the sequence." When all service times are exponential and the arrival process is Poisson, Jackson (1957) tells us each workstation works independently in steady state. System queueing time is simply the summation of the queueing times of all workstations.

The above results address queueing time, but not cycle time. Therefore, in the following study, we are going to separate queueing time from cycle time by applying decomposition to system cycle time, where system cycle time is the summation of system queueing time and system processing time. System queueing time is the summation of the queueing times of all workstations. Similarly, system processing time is the summation of all processing times, where processing time is the minimum time that a job needs in order to complete its process. When batching or assembly does not exist, processing time is the cycle time of a job in light traffic. It should be noted that the definition of processing time is different from service time, since service time is the reciprocal of capacity, which is the maximum throughput rate of a server.

This decomposition initially may seem trivial, but it is not. If all service times are constant, system queueing time is solely determined by the bottleneck workstation. Therefore, we can make the total processing time as long as we want by adding more non-bottleneck servers to the system. In this way, system cycle time can be as long as we want, but system queueing time does not change at all.

Service time is not necessary the same as processing time in practice. For example, a wet bench consists of two tanks. The first tank has constant service time of 1 minute, and the second tank has constant service time of 2 minutes. The system service time of this wet bench is 2, but the processing time is 3. In queueing theory, queueing time only relates to service time, where its mean is the reciprocal of capacity. Queueing time may not be directly associated with processing time. However, to simplify our model, our sixth assumption is to assume *processing time is the same as service time*, although it may not be true in general. However, its impact can be alleviated when utilization is high, since cycle time is dominated by queueing time in heavy traffic.

It should be noted that the above described scenario is not limited to wet benches, but also can be extended to any tool with pre-processing steps, such as a load port. You may think of the first tank in the previous example is a load port, where each load takes

one time unit. We call a machine a cascading machine, if more than one job can be processed at the same time inside the machine.

In the wet bench example, if the inter-arrival time between the first and the second jobs is 2.5 min, the machine is viewed as busy (i.e. no idle) in the first 5.5 min (= 2.5 + 1 + 2), since the second job arrives before the first job departs (i.e. smaller than 3). But, indeed, the bottleneck (i.e. the second tank) has 0.5 min of idle time during this 5.5 min. Even if the machine is kept busy, the machine is not fully utilized up to its capacity.

This gives us a hint of the complexity of the situation in practice. Keep in mind that mean service time is the reciprocal of capacity. If the service time of each tank is not constant, determining the variance of system service time from historical data is not trivial. In reality, the cascading of a machine can be more than two steps. For example, a load port plus two tanks is considered as 3 steps (i.e. 3 jobs can be processed at the same time). The job transition between two consecutive steps is executed by robots. Sometimes, completed jobs need to wait for the robot if it is busy. Furthermore, the bottleneck can be different for different products. Under the existence of complex machine configurations, robot scheduling, interruptions, resource contention, rework and product mix, finding out the variance of system service times can be intricate. Although service time SCV is well defined in theory, it may not be so accessible in practice. This could explain why we seldom see papers discussing variability of practical manufacturing systems formally. However, following the conventions, our seventh assumption is to assume that *service time SCV is available*. This strong assumption will be made in Chapter 5 and 6, but relaxed in Chapter 7.

4.3.2 Causes of Queueing Time

Queueing time is the duration that a customer waits for service. When capacity is sufficient (i.e. utilization is smaller than 1), there can be two different effects which may

cause queueing time in practical manufacturing systems: randomness effects and synchronization effects.

A randomness effect usually comes from the variation of inter-arrival times and service times. It could also be caused by interruptions or resource contentions as we have seen in Ch. 2. When randomness effect exists, based on Kingman's heavy traffic approximation, queueing time will approach infinity when the arrival rate approaches the service rate.

When there is no randomness in the system, queueing can be completely avoided if arrival intervals are synchronized with service times. For example, the inter-arrival times are constant 40, and the service times are constant 30.

However, queueing time can still occur when the arrival intervals and service times are not synchronized, even when there is no randomness in the system. This occurs when a man-made control (vs. natural behavior) is exerted on the system to achieve a pre-specified objective. For example, the inter-arrival times are always two constant 20s and then followed by one 80 (e.g. 20, 20, 80, 20, 20 and 80, etc.). Service times are 30. Although the mean inter-arrival times are 40, there are always queueing times for the last two jobs of every 3-tuple. This arrival pattern can result from the changeover rules (or, dispatching rules, in general) of the upstream machines: it may send downstream machines some jobs continuously, and then stop sending for a while. In this case, the inter-arrival times are not identically and independently distributed (i.i.d.), which violates the i.i.d. assumption of a renewal process. Because the arrival process is time-varying, it is a non-stationary process. Transfer batches (or parallel process batches) with constant batch size k are boundary cases of the above, since the inter-arrival times (or service times) can be viewed as k-1 zeros plus a large positive.

Queueing time caused by a synchronization effect is commonly seen in assembly lines. Even if all service times and inter-arrival times are deterministic, a component may still wait for other components in front of an assembly stage. It also can be induced by

shift schedules. For example, although all machines work 24 hours a day, some machines need assistance from operators. Operators only work 10 hours a day with one hour break in between (i.e. 5 + 1 + 5). Machines which need operators can keep working until finishing their current jobs even without operator. But a job has to wait if it arrives at those machines when operators are not available. When there are shift schedules, queueing time can occur even if all service times and arrival intervals are deterministic.

In practice, queueing times are caused by the mix of the randomness and synchronization effects. Usually, queueing time caused by a synchronization effect can be analyzed exactly if there is no randomness in the system. However, the exact analysis becomes difficult if it is a mix of the two. The analysis becomes extremely difficult if it is combined with the non-renewal departure process in general queueing networks.

When the arrivals follow a renewal process (which is not true in general queueing networks), even if both randomness and synchronization effects occur, some boundary cases can be analyzed exactly, such as $M/M^k/1$, $M/G^k/1$, $G^k/M/1$ and $G/M^k/m$, etc. (see Chaudhry and Templeton 1983). However, we have to resort to approximations when both inter-arrival times and service times are generally distributed (as discussed in Chapter 3). Furthermore, apart from these special boundary cases, the analysis becomes more complicated in general cases (with non-i.i.d. inter-arrival times).

For simplicity, we would first focus on the queueing times caused by the randomness effect to develop our approximate models. Therefore, our eighth assumption is to assume that *all queueing times are caused by randomness effects* in Chapter 5 and 6. However, although our models are motivated by the randomness effect, they will be tested and applied to approximate the performance of manufacturing systems subject to synchronization effect in Chapter 7.

To test the performance of the approximate models, we resort to simulations. In most cases, we assume *the service time and the initial arrival processes are gamma distributed*. This assumption will be carried on in Chapter 5 and 6 when we develop the

approximate models, but will be relaxed in Chapter 7, where the test cases come from industry. In addition to the above assumptions, we also assume that *the arrival process and the service times are mutually independent*.

4.4 Definition

Wu (2005) defines a bottleneck as "*the constraint that prevents a factory attaining its production goal(s)*". Therefore, there can be different kinds of bottlenecks based on different production goals, such as cycle time bottlenecks or throughput bottlenecks. Furthermore, the throughput bottleneck can be defined as "*the machine with the highest utilization*". We will follow the same definition in this thesis. Furthermore, when the bottleneck is mentioned without qualification, we specifically mean throughput bottleneck.

Variability (α) can be used to quantify the trade-off between queueing time and throughput. From Wu (2005), variability of a single server can be defined as

$$\alpha = \frac{E(QT)}{QT(M/M/1)} = E(QT)/\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu}. \tag{4.1}$$

Therefore, based on Heyman (1975), variability of a G/G/1 queue in heavy traffic can be approximated by

$$\alpha \cong \left(\frac{c_a^2 + c_s^2}{2}\right). \tag{4.2}$$

Variability of a factory is much more complicated and will be discussed in Chapter 7.

CHAPTER 5

BEHAVIOR OF TWO SINGLE-SERVER QUEUES IN SERIES

Logic will get you from A to B. Imagination will take you everywhere…

~ Albert Einstein

5.1 Introduction

From Chapter 4, we know that understanding dependencies among workstations in queueing networks is an important step towards understanding the behavior of practical manufacturing systems. Although the product form network theory is beautiful and powerful, its underlying assumptions seem to deviate in significant ways from the nature of practical manufacturing systems. Therefore, it may not be directly helpful in understanding the behavior of practical manufacturing systems.

In this chapter, we focus on the behavior of simple tandem queues, since they are the simplest queueing network systems which exhibit dependence. We explore the dependence inside simple tandem queues with the help of the insight from Jackson (1957) and Friedman's (1965) exact results. An interpolation/extrapolation approximation based on the ASIA (all see initial arrivals) system and Friedman's exact results is introduced. The results will be used to explain the behavior of more complex manufacturing systems in later chapters.

Due to the dependencies between nodes in general queueing networks, finding an exact solution is hard or perhaps even impossible. Therefore, all existing methods are essentially approximation methods. Some approximate models have been introduced in Chapter 4. In this chapter, we will compare the performance of our interpolation approximation with two popular existing approximate models: QNA, which is a parametric-decomposition approximation, and QNET which is based on a diffusion

approximation. We will start with the discussion of the parametric-decomposition approximation, since it gives us some useful insights towards developing our new approximate model. In understanding the parametric-decomposition approximation, Kingman's approximation and Marshall's equation play key roles.

We will first discuss Kingman's approximation in Section 5.2, and then the output processes and Marshall's equation in Section 5.3. Issues caused by the independence assumption of parametric-decomposition approximation will be analyzed in Section 5.4. The impacts on queueing time approximate error will be analyzed for two types of simple tandem queues in Section 5.5. In Section 5.6, the performance of parametric-decomposition and diffusion approximations is analyzed with simulations. The structure of simple tandem queues is introduced in Section 5.7. Approximate models for simple tandem queues are given in Section 5.8.

5.2 Kingman's Approximation

Because Kingman's approximation plays a key role in understanding the parametric-decomposition approximation, we begin by discussing its properties. As we have seen in Chapter 2, when approximating the performance of a single server, Kingman's approximation (Heyman 1975) tells us how to deal with general service times and inter-arrival times. The mean queueing time (QT) is given by the approximation

$$E(QT) = \left(\frac{c_a^2 + c_s^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu}, \qquad (5.1)$$

where μ is the maximum throughput rate (or capacity), ρ is utilization, c_a^2 and c_s^2 are the SCV of arrival intervals and service time, respectively. Mean service time is $1/\mu$. Since Kingman's approximation is motivated by heavy traffic theory, when arrivals do not follow a Poisson process, it only gives good approximations when the utilization is high. The approximation only needs the first and second moments of service times and inter-

arrival times. However, a key element behind this powerful equation is the strong assumption of identically and independently distributed (i.i.d.) arrival intervals and service times. In practice, the arrival process may not be a renewal process, especially in queueing networks representing manufacturing systems.

If inter-arrival times and service times are exponential, Kingman's approximation reduces to an M/M/1 queue and gives exact solutions. When the service times are generally distributed, Eq. (5.1) reduces to an M/G/1 queue. It still gives the exact solution and is known as the Pollaczek-Khintchin (P-K) formula (1932). Like Kingman's approximation, one important property of the M/G/1 model is that it requires only the first and second moment of the service times. In other words, when the arrival process is Poisson, for every service time distribution, the expected queueing times are the same, as long as their first and second moments are the same.

Kingman's equation is an approximation when the arrival process is not Poisson. Shanthikumar and Buzacott (1980) examined the performance of Kingman's approximation and four other G/G/1 approximate models under four different service time SCV (0.1, 0.2, 0.5, and 1) with gamma distribution. In all four cases, Kingman's approximation gives large errors for expected queue length (more than 10%) at 80% utilization when the inter-arrival time SCV is close to zero.

In order to broadly examine the performance of Kingman's approximation, we conduct five experiments for service times with a gamma distribution with SCV of 0, 0.5, 1, 2 and 10. The arrival process also follows a gamma distribution with SCV ranging from 0.1 to 8. Each data point represents the average of 100 replications. Each replication is composed of 200,000 samples after discarding the first 1,051,200 data points for warm-up. In all cases, the half-width 90% confidence intervals of queueing time are between 0.18~1.19%. Figure 5.1 shows the queueing time estimation errors under different c_s and c_a combinations at 80% utilization. The error of Kingman's approximation is positive when c_a is smaller than 1 and negative when c_a is greater than 1. The largest error is

51.2% when both service time and inter-arrival time SCVs are 0.1 (not shown in the figure).

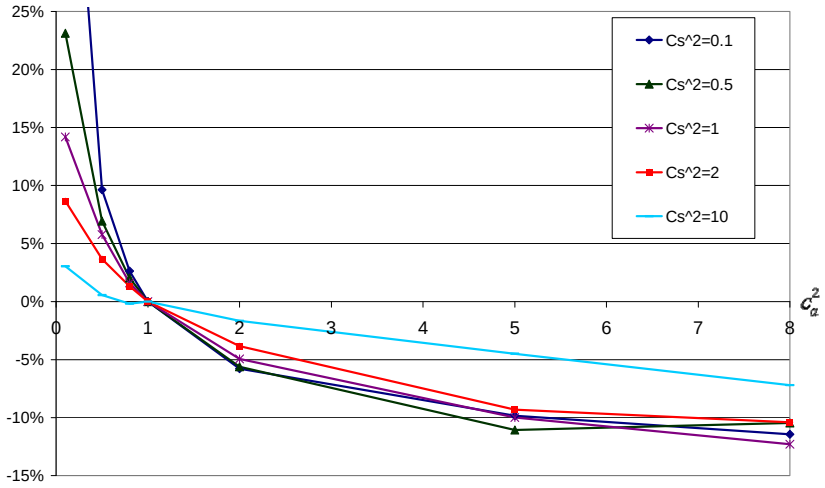

Figure 5.1 Error of Kingman's approximation at 80% utilization

By observing Figure 5.1, we find that if the arrival process is not Poisson, the absolute error of Kingman's approximation tends to increase as the service time SCV decreases, and as the inter-arrival time SCV moves away from 1. It should be noted the different roles of inter-arrival time and service time variability in Kingman's approximation: when the arrival process is Poisson, Kingman's approximation gives exact results; however, whether the service times are exponential or not does not play the same role. Indeed, service time variability will only induce errors when the arrival process is not Poisson.

Based on the Berry–Esseen theorem (1945), the distribution of sample means converge to the normal distribution slower when the variance is small. This gives us some intuition behind the large errors at small service time and inter-arrival time variabilities. Kingman's approximation is derived based on the assumption of Brownian motion, and the Brownian motion is based on the central limit theorem. If the sample

mean does not converge to normal when sample size is large (e.g. c_s = c_a = 0), the assumption of Brownian motion is not valid.

To reduce the errors caused by non-Poisson arrivals, Kraemer and Lagenbach-Belz (1976) heuristically extended Kingman's approximation based on numerical experimental results. Their simulation results show the refinement works well for both c_a smaller than 1 (based on the hypo-exponential distribution) and greater than 1 (based on the hyper-exponential distribution). While there is only one gamma distribution for a fixed set of mean and SCV, by using different parameters, we can have different hypo- and hyper-exponential distributions with the same inter-arrival time mean and SCV. Without specifying the details, Kraemer and Lagenbach-Belz's (K-L) gave the following refinement of Kingman's approximation:

$$E(QT) = g(c_a^2, c_s^2, \rho)\left(\frac{c_a^2 + c_s^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu},$$

where

$$g(c_a^2, c_s^2, \rho) = \begin{cases} \exp\left(\dfrac{-2(1-\rho)(1-c_a^2)^2}{3\rho(c_a^2 + c_s^2)}\right) & \text{if } c_a^2 < 1, \\ \exp\left(\dfrac{-(1-\rho)(c_a^2 - 1)}{c_a^2 + 4c_s^2}\right) & \text{if } c_a^2 \geq 1. \end{cases}$$

The definitions of parameters are the same as Eq. (5.1). The value of g(.) is smaller than 1 when c_a is greater than 1, which means the error of Kingman's equation is positive when c_a of a hyper-exponential distribution is greater than 1. Indeed, the errors based on hyper-exponential distributions can be either positive or negative depending on the parameter settings. However, based on the results from gamma distributions, the value of g(.) is greater than 1 when c_a is greater than 1. Therefore, for the gamma distributed service time and inter-arrival time, we have modified Kraemer and Lagenbach-Belz's (K-L) refinement as follows:

$$E(QT) = g(c_a^2, c_s^2, \rho)\left(\frac{c_a^2 + c_s^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu}, \qquad (5.2)$$

where

$$g(c_a^2, c_s^2, \rho) = \begin{cases} \exp\left(\dfrac{-2(1-\rho)(1-c_a^2)^2}{3\rho(c_a^2 + c_s^2)}\right) & \text{if } c_a^2 < 1, \\ \exp\left(\dfrac{(1-\rho)(c_a^2 - 1)}{c_a^2 + 4c_s^2}\right) & \text{if } c_a^2 \geq 1. \end{cases}$$

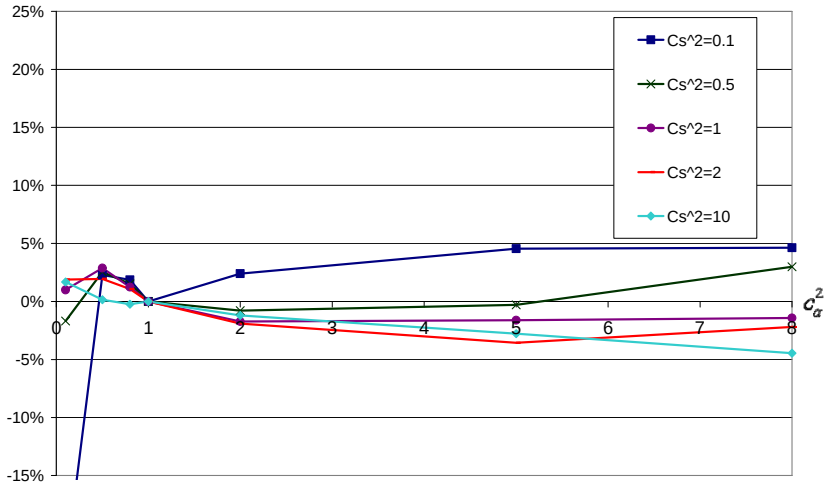

Figure 5.2 Error of K-L refinement at 80% utilization

By applying K-L refinement to the same set of data generating Figure 5.1, we get the improved errors as demonstrated in Figure 5.2. The errors become much smaller and most errors are confined within 5%. The only exception is the largest one with -23.0% when both service time and inter-arrival time SCVs are 0.1 (not shown in the figure). However, comparing to the original 51.2% error, it is still an impressive improvement. Furthermore, when both SCVs are small (i.e. 0.1), the queueing time is relatively short. Large relative error in queueing time may only contribute relatively small error to the cycle time estimate. The improvement is more significant when SCVs are large. The K-L refinement seems to perform well in this region.

The K-L refinement is an important result, since it reduces the errors caused by non-Poisson arrivals. It can be used to improve the queueing time approximation of the servers in the ASIA system. However, it should be noted that the modification is motivated by the observations from gamma distributions only.

Notice that Eq. (5.2) reduces to Eq. (5.1) when ρ is 1, c_a is 1 or c_s goes to infinity. Specifically, the value of g(.) is away from 1 when utilization is lower, which implies the errors of Eq. (5.1) become larger in light traffic.

In summary, Kingman's formula gives us exact solutions when the arrival process is Poisson. If arrival process is not Poisson, it still approaches the exact value in heavy traffic, or when c_s goes to infinity. However, based on our second assumption stated in Chapter 4, c_s in practical manufacturing systems would be expected to be small. Therefore, under the assumption that service time is i.i.d., Kingman's formula gives us unreliable estimates when the arrival process is renewal, but not Poisson and traffic intensity is low. However, due to the i.i.d assumption, one can also expect that the solution is unreliable if the arrival process is not renewal. In summary, Kingman's formula gives us unreliable solutions under the following two conditions:

1. Arrival process is renewal, but not Poisson and traffic intensity is low, or
2. Arrival process is not renewal.

5.3 Output processes and Marshall's Equation

For the output process of a G/G/1 queue with renewal arrival process, Marshall (1968) shows the variance is

$$Var(D) = \sigma_a^2 + 2\sigma_s^2 - (2/\lambda)(1-\rho)E(QT), \qquad (5.4)$$

where D is inter-departure time between two subsequent departures, σ_a is the standard deviation of inter-arrival time, σ_s is the standard deviation of service time. It can be transformed into the following,

$$c_d^2 = c_a^2 + 2\rho^2 c_s^2 - (2\rho)(1-\rho)\mu E(QT), \tag{5.5}$$

where μ is the service rate (or capacity), c_d is the coefficient of variation of the departure interval. The other variables are as defined in Eq. (5.1). If we substitute Kingman's approximation into Eq. (5.5), we get the following simple formula

$$c_d^2 = \rho^2 c_s^2 + (1-\rho^2)c_a^2. \tag{5.6}$$

Eq. (5.6) has a nice intuitive explanation. For a single server, the SCV of its departure process is the convex combination of its service time SCV and its inter-arrival time SCV. When queueing jobs almost always exist (utilization is high), SCV of inter-departure time is dominated by the SCV of service time. When the queue is almost always empty (utilization is low), the SCV of inter-departure time is dominated by the inter-arrival time SCV (Hopp and Spearman 1996).

Because Eq. (5.6) is derived based on Kingman's approximation, it also inherits the limitation of Kingman's equation. It gives us exact results when the arrival process is Poisson and service time is i.i.d. Otherwise, it is only an approximation. If the service time and inter-arrival time follow the gamma distribution, when c_a is smaller than 1, Kingman's approximation tends to overestimates the queueing time. Marshall's equation tends to underestimate c_d, which can be seen from Eq. (5.5). The true c_d can be larger than the approximation (or closer to 1). On the other hand, when c_a is greater than 1, Kingman's approximation tends to underestimates the queueing time. Marshall's equation tends to overestimate c_d. The true c_d can be smaller than the approximation (or closer to 1). It should be noted the true value is always closer to 1 than the approximation.

5.4 Issues with the Parametric-Decomposition Approximation

In practical manufacturing systems, there is dependence among workstations. However, if we ignore the dependence, the analysis is much simpler and more tractable. This approach is called parametric-decomposition approximation, which it is indeed the core idea of Queueing Network Analyzer (QNA) proposed by Whitt (1983).

After examining the properties of Kingman and Marshall's equations, we are ready to look at the queueing time of the second server in simple tandem queues using the parametric-decomposition approach. We can have the arrival process SCV of the second server by Marshall's equation, and apply Kingman's approximation to get the queueing time of the second server. To analyze the error caused by this approach, we need to come back to the two sources of error in Kingman's approximation discussed in Section 5.2.

The first error can be alleviated by the K-L refinement. The challenge comes from the second, which does not occur at the first server, since the initial arrival process is usually assumed to be renewal. But the arrival process is not in general renewal for the second server. How much error will this cause? This question will be investigated using simulations in Section 5.6.

Before conducting simulations, we first want to compare the queueing time approximation error from two types of simple tandem queues. The queueing time approximation error of the second queue may or may not have serious impact on the system cycle time approximation.

5.5 Simple Tandem Queues with Front-End or Backend Bottlenecks

If the initial arrival process is renewal, we know from Section 5.2 that the queueing time of a single server can be approximated by Kingman's approximation. Therefore, approximating the queueing time for the first server in simple tandem queues

is usually not an issue. The challenge comes from the second server. Because the departure process of the first server is typically not renewal, we need a new method to accurately analyze the queueing time for the second server.

In simple tandem queues, there can be one of three situations: the first server has higher utilization, the second server has higher utilization, or both utilizations are the same. We call the first situation simple tandem queues with front end bottleneck (STQF), and the second situation simple tandem queues with backend bottlenecks (STQB). In STQF, the queueing time of the second server is relatively shorter than the queueing time of the first server, especially at high utilizations. Large approximation errors at the second server only contribute small errors to the total system queueing time. On the other hand, in STQB, the system queueing time is dominated by the second server especially at high utilizations. For example, in a simple tandem queue with Poisson arrival and exponential service time, the queueing time ratio (QTR) is

$$QTR = \frac{E(QT_1)}{E(QT_2)} = \frac{\rho_1/(1-\rho_1)}{\rho_2/(1-\rho_2)} \frac{1/\mu_1}{1/\mu_2} = \frac{\mu_2(\mu_2-\lambda)}{\mu_1(\mu_1-\lambda)}. \quad (5.7)$$

If μ_1 is 3, μ_2 is 1, and λ is 0.9, the ratio is 1/63. Indeed, this ratio goes to zero, when λ approaches μ_2. Figure 5.3 gives an example to show how fast the queueing time ratio decreases, when service times of the first and second server are 25 and 30 in an M/M/1 $\rightarrow$ M/1 system. The two increasing curves are the queueing times of the first and second server. The decreasing curve is the queueing time ratio from Eq. (5.7). The x-axis is the utilization of the second server, the y-axis on the left is the queueing time and the y-axis on the right is queueing time ratio.

Based on the parametric-decomposition approach and Kingman's approximation, the queueing time is determined by both the variability term and the utilization term, where the impact from the variability term is linear but the impact from the utilization term is exponential. Since Eq. (5.7) only captures the ratio of the utilization terms in a simple tandem queue, the true ratio should be scaled by the ratio of the variability terms.

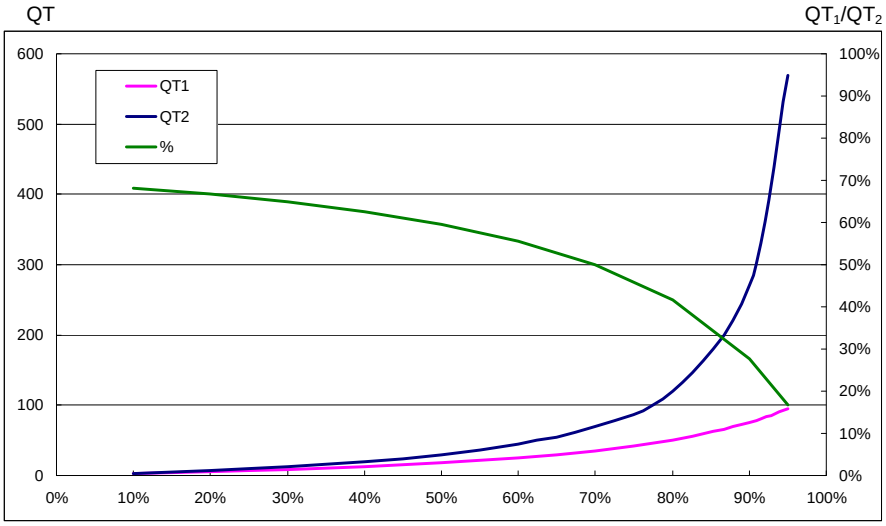

Figure 5.3 Queueing time ratio in STQB

In the cases of STQB, because the QTR is small and decreases in heavy traffic, the second queueing time dominates the system queueing time in heavy traffic. Any improvement on the second queueing time approximation will contribute nearly the same amount of improvement on the system queueing time estimate, where system queueing time is the summation of the two queueing times. Therefore, approximating the second queueing time accurately is important in STQB.

On the other hand, when the first server is the bottleneck, the QTR can be very large. The second queueing time only contributes to a small portion of the system queueing time in heavy traffic. The approximate error of the second queueing time plays a less important role in terms of system queueing time. More explanation on STQF will be given in Section 5.8.3.

When there are two bottlenecks, from Marshall's and Kingman's equations, we can expect that the two queueing times are roughly of the same order. From simulations, if the arrival process is Poisson and service time variability is smaller than 1, the ratio

ranges from 0.67 to infinity, depending on their specific variabilities. Because the ratio is not small, this scenario is combined into STQF.

5.6 Performance of Parametric-Decomposition and Diffusion Approximations

The two approximate models, QNA and QNET, have been introduced in Chapter 4. In this section, we are going to test their performance on simple tandem queues. Since a general queueing network can be viewed as an aggregation of simple tandem queues, if an approximate model causes large error in simple tandem queues, it might not be reasonable to expect it performs well in general queueing networks.

The QNA approximation is obtained by combining Marshall and Kingman's equations as introduced before. The approximation from QNET is obtained by running the QNET code developed by Dai (1992).

In some special cases, the true performance can be obtained analytically. However, in most cases, we need to conduct simulation experiments to get the sample means and corresponding confidence intervals. Based on the discussion in Section 5.5, the experiments are separated into two groups: STQF and STQB. As we have explained, the error in STQB has more significant impact on the system cycle time approximation.

A simple tandem queue can be analyzed exactly in the following two situations: (1) both service times are constant, or (2) initial arrival process is Poisson and the service time of the first server is exponential. In the first case, is the result is obtained by the reduction method, where the system queueing time is simply determined by the bottleneck workstation. In a simple tandem queue with constant service times, if the first server is the bottleneck, based on the reduction method, the queueing times are

$$QT_1 = \left(\frac{c_{a1}^2 + 0}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \tag{5.8}$$

$$QT_2 = 0. \tag{5.9}$$

If the second server is the bottleneck, the queueing times become

$$QT_1 = \left(\frac{c_{a1}^2 + 0}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \tag{5.10}$$

$$QT_2 = \left(\frac{c_{a1}^2 + 0}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - \left(\frac{c_{a1}^2 + 0}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}. \tag{5.11}$$

The second case is the result of Burke's theorem (1956). When the initial arrival process is Poisson and the service time of the first server is exponential, the second server also sees Poisson arrivals and can be analyzed exactly by an M/G/1 queue. If the second service time is also exponential, this simple tandem queue becomes a Jackson network.

Table 5.1 shows the four examined STQB cases: (1, 1, 0), (1, 0, 1), (1, 0, 0) and (1, 1, 1) where the values in each bracket are the arrival process SCV, service time SCVs of the first and second servers, respectively. All the intervals follow a gamma distribution. Therefore, the arrival process is Poisson. When the service time SCV is 1, it the distribution is exponential.

In each SCV combination, the service time of the second server is always 30, but the service time of the first server is 10, 20, 25 or 29. We do not take the value of 30, because we want a distinct bottleneck. We do not take equal distances (i.e. 10, 20 and 30) because in practical manufacturing systems, machine utilizations are expected to be high and thus, close to bottleneck utilization. Therefore, we add a 25 between 20 and 29 to reflect this tendency. The utilization of the bottleneck ranges between 10% and 95%. The four blocks from top down are the queueing times of the first and second servers, error % of QNA, and error % of QNET, respectively.

Except for the case of (1, 0, 1), the other three cases fall under the above two conditions. Therefore, the queueing time can be determined analytically. Only for the (1, 0, 1) case, do we need to resort to simulation. In that case, each observation is the average of 100 replications. Each replication is the average of 200,000 ~ 1,000,000 data points after a warm up period of 50 ~ 65 years (i.e. 87,600 ~ 1,081,860 data points are

discarded). The largest half-width 90% confidence interval of queueing time is 1.61%, and most of them are smaller than 0.5% except for some high utilization cases.

The error percentage is "(approximation value – reference result) / reference result", where the reference result is either the exact analytical result or simulation result. While QNA gives the exact results, and QNET approximate very closely the exact results under the second condition (i.e. c_{e1} is 1), they both perform terribly in the first situation (i.e. c_{e1} and c_{e2} are 0), where the queueing time can be easily calculated based on the reduction method. In this situation, QNET reports that all the second station queueing times are very close to 0 for the (1, 0, 0) case. Indeed, it can be as large as 285 (i.e. 10/30 at 95%). While QNA may have large errors (i.e. 1335.66%) at low utilization, the impact to system cycle time error in this case is small, since system service time is relatively long (i.e. 59) compared with the second queueing time (i.e. 0.1).

QNET is developed based on a Brownian approximation, and the Brownian motion is based on the central limit theorem. The central limit theorem states that the sum of a sufficiently large number of independent random variables will be approximately normally distributed. The key is, how large is sufficiently large? If the random variable is a constant with zero variance, no matter how many constants we collect, their distribution will not converge to a normal distribution (the Berry–Esseen bound (1945) becomes infinity). This explains the failure of QNET, which gives -100% errors, when both service times are constant. Therefore, in order to make QNET work, we need some randomness in the service time distribution.

Table 5.1 Performances of QNA and QNET

(C_{a1}^2, C_{a1}^2, C_{s2}^2)	Sim QT of The 1st Server				Sim QT of The 2nd Server				Error % of QNA				Error % of QNET			
BN Util \ (ST1/ST2)	Exp-Exp-Const (1, 1, 0)				Exp-Const-Exp (1, 0, 1)				Exp-Const-Const (1, 0, 0)				Exp-Exp-Exp (1, 1, 1)			
	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30
10%	0.3	1.4	2.3	3.1	0.2	0.7	1.1	1.6	0.2	0.7	1.1	1.6	0.3	1.4	2.3	3.1
20%	0.7	3.1	5.0	7.0	0.4	1.5	2.5	3.5	0.4	1.5	2.5	3.5	0.7	3.1	5.0	7.0
30%	1.1	5.0	8.3	11.8	0.6	2.5	4.2	5.9	0.6	2.5	4.2	5.9	1.1	5.0	8.3	11.8
40%	1.5	7.3	12.5	18.3	0.8	3.6	6.2	9.1	0.8	3.6	6.3	9.1	1.5	7.3	12.5	18.3
50%	2.0	10.0	17.9	27.1	1.0	5.0	8.9	13.6	1.0	5.0	8.9	13.6	2.0	10.0	17.9	27.1
60%	2.5	13.3	25.0	40.0	1.2	6.7	12.5	20.0	1.3	6.7	12.5	20.0	2.5	13.3	25.0	40.0
70%	3.0	17.5	35.0	60.7	1.5	8.7	17.5	30.3	1.5	8.8	17.5	30.3	3.0	17.5	35.0	60.7
80%	3.6	22.9	50.0	98.9	1.8	11.4	25.0	49.5	1.8	11.4	25.0	49.5	3.6	22.9	50.0	98.9
90%	4.3	30.0	75.0	194.1	2.1	15.0	37.5	97.3	2.1	15.0	37.5	97.0	4.3	30.0	75.0	194.1
95%	4.6	34.5	95.0	326.1	2.3	17.3	47.5	162.6	2.3	17.3	47.5	163.1	4.6	34.5	95.0	326.1
10%	1.7	1.7	1.7	1.7	3.2	2.9	2.7	2.5	1.5	1.0	0.5	0.1	3.3	3.3	3.3	3.3
20%	3.8	3.8	3.8	3.8	7.2	6.5	6.0	5.6	3.4	2.2	1.3	0.3	7.5	7.5	7.5	7.5
30%	6.4	6.4	6.4	6.4	12.4	11.2	10.4	9.7	5.9	3.9	2.3	0.5	12.9	12.9	12.9	12.9
40%	10.0	10.0	10.0	10.0	19.4	17.6	16.3	15.2	9.2	6.4	3.8	0.9	20.0	20.0	20.0	20.0
50%	15.0	15.0	15.0	15.0	29.2	26.7	24.7	22.8	14.0	10.0	6.1	1.4	30.0	30.0	30.0	30.0
60%	22.5	22.5	22.5	22.5	43.9	40.6	37.7	34.4	21.3	15.8	10.0	2.5	45.0	45.0	45.0	45.0
70%	35.0	35.0	35.0	35.0	68.8	64.1	59.4	53.9	33.5	26.3	17.5	4.7	70.0	70.0	70.0	70.0
80%	60.0	60.0	60.0	60.0	118.6	112.2	104.9	93.8	58.2	48.6	35.0	10.5	120.0	120.0	120.0	120.0
90%	135.0	135.0	135.0	135.0	268.5	260.2	245.1	218.5	132.9	120.0	97.5	38.0	270.0	270.0	270.0	270.0
95%	285.0	285.0	285.0	285.0	570.9	558.2	537.6	473.0	282.7	267.7	237.5	121.9	570.0	570.0	570.0	570.0
10%	0.00%	0.00%	0.00%	0.00%	4.32%	16.00%	24.47%	32.63%	11.41%	74.22%	212.10%	1335.66%	0.00%	0.00%	0.00%	0.00%
20%	0.00%	0.00%	0.00%	0.00%	3.88%	14.52%	22.34%	30.57%	10.04%	66.55%	191.67%	1213.65%	0.00%	0.00%	0.00%	0.00%
30%	0.00%	0.00%	0.00%	0.00%	3.07%	12.18%	19.41%	26.94%	8.36%	57.09%	166.45%	1063.54%	0.00%	0.00%	0.00%	0.00%
40%	0.00%	0.00%	0.00%	0.00%	2.31%	9.52%	15.77%	22.05%	6.41%	45.97%	137.04%	890.44%	0.00%	0.00%	0.00%	0.00%
50%	0.00%	0.00%	0.00%	0.00%	1.28%	6.23%	11.00%	16.14%	4.17%	33.33%	104.17%	700.83%	0.00%	0.00%	0.00%	0.00%
60%	0.00%	0.00%	0.00%	0.00%	0.36%	2.01%	4.55%	8.68%	1.65%	19.37%	68.75%	502.98%	0.00%	0.00%	0.00%	0.00%
70%	0.00%	0.00%	0.00%	0.00%	-0.98%	-2.75%	-2.28%	0.20%	-1.15%	4.30%	31.94%	307.64%	0.00%	0.00%	0.00%	0.00%
80%	0.00%	0.00%	0.00%	0.00%	-2.45%	-8.29%	-11.01%	-10.28%	-4.21%	-11.61%	-4.76%	129.05%	0.00%	0.00%	0.00%	0.00%
90%	0.00%	0.00%	0.00%	0.00%	-3.96%	-14.93%	-20.83%	-23.19%	-7.53%	-28.00%	-39.42%	-13.55%	0.00%	0.00%	0.00%	0.00%
95%	0.00%	0.00%	0.00%	0.00%	-5.16%	-18.37%	-27.20%	-30.31%	-9.29%	-36.25%	-55.21%	-63.39%	0.00%	0.00%	0.00%	0.00%
10%	0.00%	0.00%	0.00%	0.00%	-21.43%	-10.39%	-3.56%	1.45%	-100.00%	-100.00%	-100.00%	-99.30%	0.00%	0.00%	0.00%	0.00%
20%	0.00%	0.00%	0.00%	0.00%	-21.16%	-11.20%	-4.09%	1.39%	-100.00%	-100.00%	-100.00%	-99.75%	0.00%	0.00%	0.00%	0.00%
30%	0.00%	0.00%	0.00%	0.00%	-20.86%	-12.39%	-4.60%	1.10%	-100.00%	-100.00%	-100.00%	-99.88%	0.00%	0.00%	0.00%	0.00%
40%	0.00%	0.00%	0.00%	0.00%	-20.12%	-13.57%	-5.06%	0.80%	-100.00%	-100.00%	-100.00%	-99.99%	0.00%	0.00%	0.00%	0.00%
50%	0.00%	0.00%	0.00%	0.00%	-19.10%	-14.97%	-5.88%	0.71%	-100.00%	-100.00%	-100.00%	-100.01%	0.00%	0.00%	0.00%	0.00%
60%	0.00%	0.00%	0.00%	0.00%	-17.34%	-16.76%	-7.72%	0.38%	-100.00%	-100.00%	-100.00%	-100.01%	0.00%	0.00%	0.00%	0.00%
70%	0.00%	0.00%	0.00%	0.00%	-15.12%	-18.36%	-9.68%	0.33%	-100.00%	-100.00%	-100.00%	-100.01%	0.00%	0.00%	0.00%	0.00%
80%	0.00%	0.00%	0.00%	0.00%	-11.95%	-19.04%	-13.50%	-0.30%	-100.00%	-100.00%	-100.00%	-100.00%	0.00%	0.00%	0.00%	0.00%
90%	0.00%	0.00%	0.00%	0.00%	-7.28%	-16.26%	-17.38%	-1.93%	-100.00%	-100.00%	-100.00%	-100.00%	0.00%	0.00%	0.00%	0.00%
95%	0.02%	-0.01%	0.01%	-0.01%	-4.49%	-11.12%	-16.47%	-3.82%	-100.00%	-100.00%	-100.00%	-100.00%	0.02%	-0.01%	0.01%	-0.01%

Since the errors are too big (-100% and 1335%, etc.) and constant service times may not be practical, new experiments are conducted for STQB where service time SCVs are chosen from 0.1 (low), 0.5 (medium), and 0.9 (high), resulting in nine experiments, i.e. (1, 0.1, 0.1), (1, 0.1, 0.5), (1, 0.1, 0.9), (1, 0.5, 0.1), (1, 0.5, 0.5), (1, 0.5, 0.9), (1, 0.9, 0.1), (1, 0.9, 0.5), and (1, 0.9, 0.9). The service times follow gamma distribution and the SCVs equally spread out between 0 and 1. The arrival process is assumed to be Poisson (i.e. SCV = 1). However, this assumption will be relaxed in Chapter 6. Each observation is the average of 100 ~ 200 replications. Depending on the utilization and the first service time, each replication is the average of 200,000 ~ 1,000,000 data points after a warm up period of 50 ~ 65 years (87,600 ~ 1,081,860 data points). The results are shown in Appendix D. For all 9 cases, the largest half-width 90% confidence interval of queueing time is 1.69%, and most are smaller than 0.5%.

The average error is 11.1% from QNA and is 13.1% from QNET. Since when the second station queueing time is small relative to the first queueing time, the error has less impact on the system queueing time approximation, it is reasonable to take the ratio of the second station queueing time to the system queueing time into account. From the table in Appendix D, we see the ratio becomes larger when service time of the first server (i.e. the first service time) is shorter. The weighted average error (i.e. error % of the 2^{nd} QT * 2^{nd} QT / (1^{st} QT + 2^{nd} QT)) from QNA is 6.3%, while QNET is 10.3%. Comparing those two errors, we find QNA seems to work better in those regions which have higher impact on system queueing times.

By observing the errors, we find QNA works poorly in the following three cases: (1) the first service time is long, (2) light traffic, or (3) heavy traffic. The third condition has been observed by Whitt (1985) who wrote: “This approximation usually performs well, but it can perform poorly under certain heavy traffic conditions...” Our results further show that QNA’s performance deteriorates in heavy traffic in all examined cases, when service time SCV is smaller than 1 with Poisson arrivals. The performance is

especially poor when the service time SCV is close to 0. The impact of small variability has also been observed by Whitt (1985), who said, “The approximation also tends to perform poorly when several consecutive stations have deterministic or nearly deterministic service-time distributions. Then our approximate decoupling of the stations tends to be unjustified.” He called the above situation the “pipelining effect”.

Indeed, the above three conditions are mainly attributed to one reason: the non-renewal departure process of the first server, or the so-called dependence between the first and the second server. As we mentioned, the non-renewal arrival process of the second server is the second error condition of Kingman’s approximation, which prevents it from giving the correct answers, even if Marshall’s equation gives exact answers under this condition (because of the initial Poisson arrivals).

In addition to the above-mentioned errors, it is important to note that QNA works well in general when system utilization is around 70 ~ 80%. Specifically, in STQB, it works well at 80% utilization when the first station service time is close to the bottleneck service time (e.g. service time is 25 or 29), and it works well at 70% utilization when the first service time is shorter (e.g. service time is 10 or 20).

Observation 5.1:

When the initial arrival process is Poisson and service time SCV is smaller than 1, the parametric-decomposition approximation performs well when the bottleneck traffic intensity is around 70% ~ 80% in simple tandem queues with backend bottlenecks.

By observing the errors in Appendix D, we find QNET performs relatively poorly in the following three situations: (1) the first service time is short, (2) light traffic, or (3) heavy traffic in some cases. The first two conditions are expected, since QNET is a diffusion approximation, which works well when both servers are heavily loaded. The most interesting aspect is the third one: QNET does not perform well in heavy traffic in

many cases (e.g. the cases of (1, 0.1, 0.1)). The tendency (i.e. performing poorly in heavy traffic) becomes stronger when the service time SCV is close to zero. As we mentioned earlier, a potential cause of this failure can be attributed to the convergence speed of the central limit theorem.

5.7 Fully Coupled System, ASIA System and Their Difference

In this section, we give several definitions, which will be used later to derive a new approximation. We start by defining a fully coupled system. Based on Friedman's reduction method (1965), system queueing time is solely determined by the bottleneck when all service times are deterministic. Inspired by this, we defined a fully coupled system as:

Definition 5.1 (Fully Coupled System):
System queueing time is determined solely by its bottleneck workstation and is the same as the bottleneck workstation queueing time would be if the bottleneck workstation sees the initial arrival process directly.

In a fully coupled system, the queueing time of any customer equals the time the same customer would have been waiting in the queue of a single workstation, which is the bottleneck workstation. Since Friedman and Avi-Itzhak's (1965) results are applicable to a tandem queue with any specified arrival process, it should be noted that Definition 5.1 is applicable to both renewal and non-renewal arrival processes.

In Kingman's approximation, the queueing time is zero if there is no randomness in the system (i.e. both service time and inter-arrival time are deterministic). Furthermore, when all service times are deterministic, there is no randomness from service times. Queueing time is merely induced by the randomness of inter-arrival time. When the

service time is not constant, as long as service times are i.i.d., it is reasonable to conjecture that we should have a longer queueing time when we add more randomness to the system by increasing service time SCV from zero to a positive value. We also expect the queueing time to be increasing in the service time SCV (i.e. variability). Without giving a rigorous proof, we propose the following conjecture:

Conjecture 5.1a (Lower Bound of Queueing Time in Tandem Queues):
For any tandem queue system with a specified arrival process, if the service time SCV of all servers gradually decreases to zero, system queueing time decreases to the queueing time in its fully coupled system.

Conjecture 5.1a tells us the queueing time given by the reduction method of Friedman is a lower bound for tandem queues, i.e., the system queueing time is no shorter than the queueing time of its corresponding fully coupled system. This conjecture will be tested by a large simulation study in Section 5.8. On the other hand, the queueing time will become longer if service time SCV increases. This tendency can be observed from Kingman's equation: if we increase the value of c_s, the queueing time always increases. However, Kingman's equation is only an approximation. In addition to the fully coupled system, an ASIA system is defined as:

Definition 5.2 (ASIA System):
For the workstations in the queueing network, all see initial arrivals (ASIA) directly.

Definition 5.2a (ASIA Tandem Queues):
In an ASIA tandem queue, all servers see the initial arrival process directly.

In an ASIA system, you may imagine that the tandem queues are re-arranged in parallel instead of serial, and each arrival generates multiple jobs. Each job is served by one of the parallel queues. If reentry exists, all the servers in this reentrant process will see the arrival process for this reentrant loop instead of the initial arrival process. It should be noted that the ASIA system queueing time is equal to or greater than the fully coupled system queueing time.

Corollary 5.1:

The ASIA system queueing time is equal to or greater than the fully coupled system queueing time.

Proof: Since the fully coupled system queueing time is the same as the bottleneck queueing time when it sees the initial arrival process, the ASIA system queueing time is the summation of the fully coupled system time plus the queueing times at the non-bottlenecks, when they see the initial arrival process directly. Q.E.D.

An ASIA system is more general than an independent system. For example, in a Jackson network, all servers work as if they are independent in steady states. Jackson networks, in some sense, are more limited than Definition 5.2, since they require exponential service times and inter-arrival times. However, a Jackson network has the same steady state distribution as an ASIA system.

For a tandem queue, if the initial inter-arrival times and all service times are exponential, all servers see the Poisson arrival process. Therefore, a Jackson type tandem queue is a special case of an ASIA system.

When service time is exponential (but the arrival process may not be Poisson), the system characteristics are given by Weber (1979). The interchangeability theorem, proved by Weber, says that for tandem queues with exponential service times, the output

process has the same distribution no matter what the sequence of the servers is. It implies that, for any ordering of the servers, the system queueing times are the same. Daley (1968) proved that "The output process of a stationary GI/M/1 queueing system is a renewal process if and only if the input process is a Poisson process, in which case the output process is a Poisson process." If the initial arrival process is renewal, the only way to make a Weber system an ASIA system is for it to have Poisson arrivals, which indeed makes it a Jackson network. Based on Definition 5.2a, we have the following conjecture:

Conjecture 5.2a (Upper Bound of Queueing Time in Simple Tandem Queues):
If the service time SCV of the first server is smaller than 1, and the initial arrival process is Poisson, the upper bound on system queueing time of a simple tandem queue is the queueing time in its ASIA system.

When the first service time is exponential, the upper bound is tight. In this situation, the second station queueing time can be modeled by the M/G/1 queue and is the same as its ASIA system queueing time. Tembe and Wolff (1974) determined an upper bound on the cycle time of the second server in an M/D/1 → M/1 system as

$$E(CT_2) \le \frac{1}{\mu_2 - \lambda} = \left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \frac{1}{\mu_2}, \qquad (5.12)$$

This bound is simply the M/M/1 cycle time of the second server, and is indeed the same as Conjecture 5.2a. They also commented that "Intuitively, constant service at station 1 seems to regularize the departure stream."

Conjecture 5.2a will be tested experimentally by simulation in Section 5.8. When the arrival process is not Poisson, we are not sure if Conjecture 5.2a still holds. However, Niu (1980) claimed similar upper bound for general arrival processes by the following conjecture: In a GI/D/1 → G/1 system, the stationary expected delay in front of the second server is smaller than it would be if there was no first server at all. This is exactly

the same as our conjecture for the upper bound of a GI/D/1 → G/1 system except that the arrival process is generally distributed. In other words, Niu conjectured that the upper bound based on Conjecture 5.2a is still exact even for general arrival process as long as the first service time is constant.

Niu also gave a looser upper bound for a stationary GI/D/1 → G/1 system,

$$E(QT_2) \le \left(\frac{\sigma_a^2 \mu_2^2 + c_{s2}^2}{2} \right) \left(\frac{\rho_2}{1-\rho_2} \right) \frac{1}{\mu_2}, \tag{5.13}$$

where σ_a is standard deviation of the arrival process, μ_2 is the mean service rate of the second server and c_{s2} is the service time SCV of the second server. Although the arrival process can be general, the service time must be deterministic. Furthermore, μ_2 must be greater than λ for system stability. This bound is looser than Conjecture 5.2a.

Based on Conjecture 5.1a and 5.2a, for simple tandem queues, if the arrival process is Poisson and service time variability is smaller than or equal to 1, the upper bound is the queueing time of its corresponding ASIA system, and the lower bound is the queueing time of its corresponding fully coupled system.

The above conjectures deal with the behavior of the second server's queueing time in simple tandem queues, when service time variability is smaller than 1, with Poisson arrivals. However, when service time variability is greater than one, the situation will be different.

Conjecture 5.2b (Lower Bound of Queueing Time in Simple Tandem Queues):
When the service time variability of the first server is greater than 1, with Poisson arrivals, the lower bound on system queueing time of a simple tandem queue is the queueing time in its ASIA system.

When the service time variability of the first server is greater than 1 and the arrival process is Poisson, Conjecture 5.2b gives a higher lower bound (i.e. its ASIA system) than the conservative bound given in Conjecture 5.1a (i.e. its fully coupled system) for a simple tandem queue. When the first service time is exponential, the lower bound is tight. Conjecture 5.2b will be tested by simulations in Section 5.8.

A more profound assumption inherent in the above conjectures is that more randomness induces longer queueing time. Therefore, under the same arrival and service time distributions (e.g. gamma), larger SCVs of the arrival process or service times will always induce longer queueing time. Indeed, all of the above conjectures are the direct results of this assumption.

Since the first station queueing time can be obtained exactly by P-K formula when the arrival process is Poisson, all of the above conjectures are not only bounds for the system queueing time but also bounds for the second station queueing time in simple tandem queues. An important observation from the above conjectures is that they seem to have nice structure embedded in a queueing network arising from its associated ASIA system (Definition 5.2) and fully coupled system (Definition 5.1). Indeed, the next thing we explore is the property of the intrinsic gap, which is defined as:

Definition 5.3 (Intrinsic Gap):
The difference in queueing times between the ASIA and fully coupled system models.

It should be noted that the construction of Definition 5.3 only depends on Definition 5.1 and 5.2. The above conjectures can be used to gain insights of the intrinsic gap, but has no direct relation with the definition. Similar (but not the same) observation of the intrinsic gap has been made by Whitt (1985): in simple tandem queues, "the approximate relative difference when $c_{s1}^2 = c_{s2}^2 = 0$ or when $c_{s1}^2 = c_{s2}^2 = 1$ should be viewed as a measure of approximation error."

In the next section, we will explain how to approximate the queueing time of simple tandem queues by using this gap as a transformer.

5.8 Approximate Models for Simple Tandem Queues

In Section 5.6, we showed that the average error of QNA is ~ 13% and the average error of QNET is ~ 16% at high utilization in STQB. The error becomes larger when service time SCV is close to 0. However, in practical manufacturing systems, the service time SCV is expected to be low (to achieve high efficiency). Both QNA and QNET perform poorly in this range. We need a better approximate model for practical manufacturing systems. In this section, we present the main result of this chapter: an interpolation approximation for the mean system queueing time in a simple tandem queue.

Since Kingman's approximation assumes the arrival process is renewal, when the arrival process is not renewal, the queueing time analysis of the second server becomes difficult. Newell (1979) commented, "… the detailed probability structure of the output is usually so complicated that, even one know it, one could not make much progress in analyzing any subsequent queues for which this might be the input." Therefore, it would be important to find if we can approximate the queueing time of the second server without using the information of the departure process. This idea indeed has been adopted by QNET. Unfortunately, QNET does not perform well in the examined cases.

An approach for dealing with the non-renewal departure process is to treat tandem queues as a whole, without making decomposition between them. In order to do this, we will start from identifying the underlying structure of tandem queues based on the observations in Section 5.7. By taking advantage of the structure, we will construct simple and more accurate approximations without dealing with the departure process.

For the new approximate models of simple tandem queues, we start with Poisson arrivals, since Kingman's equation gives exact solutions under this situation. Other

general renewal arrival process can be considered by extending this model, with extra errors introduced by Kingman's approximation. We introduce the model for STQB first.

5.8.1 Structure of STQB with Small Service Time Variability

When service time variability is between zero and one, based on Conjecture 5.2a, the upper bound of the second station queueing time (QT_2^U) in simple tandem queues is given by its ASIA system. Therefore,

$$QT_2^U = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}, \tag{5.14}$$

where c_{s2}^2 is the service time SCV of the second server, μ_2 is the mean service rate (i.e. capacity) of the second server, and ρ_2 is the utilization of the second server.

When service time variability is between zero and one, based on Conjecture 5.1a, the lower bound of the second queueing time in STQB is given by its fully coupled system. Therefore,

$$QT_2^L = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - \left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \tag{5.15}$$

where c_{s1}^2 is the service time SCV of the first server, μ_1 is the mean service rate of the first server, and ρ_1 is the utilization of the first server. The rest is the same as above. The intrinsic gap (IG) for STQB is

$$IG = QT_2^U - QT_2^L = QT_1 = \left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}. \tag{5.16}$$

In STQB, the intrinsic gap is exactly the same as the queueing time of the first server! Therefore, this intrinsic gap possesses the nice property we have demonstrated by Eq. (5.7): When system utilization approaches 1, the ratio of the intrinsic gap to the second station queueing time approaches zero.

Property 5.1 (Heavy Traffic Property of the Intrinsic Gap for STQB):
In simple tandem queues with backend bottlenecks, the ratio of the intrinsic gap to the queueing time of the bottleneck server goes to zero as the traffic intensity approaches 1.

If the traffic intensity of the second server approaches 1, the ratio of intrinsic gap goes to zero and the conjectured lower bound approaches the conjectured upper bound. If the true queueing time is between the lower and upper bound, the second queueing time approaches the queueing time in its ASIA system (i.e. upper bound). This result is consistent with the heavy traffic bottleneck phenomenon observed by Suresh and Whitt (1990b) and Whitt (2003). It says when the bottleneck utilization approaches 1, the bottleneck queueing time distribution is asymptotically the same as if the immediate arrival process were replaced by the external arrival process to the first server.

By the previous conjectures, Eq. (5.14) and (5.15) give us an upper bound and lower bound, respectively. It is important to verify if those bounds are valid, and how the true bottleneck queueing time behaves between these two bounds. An empirical validation is given by the simulation results in Appendices C and D. The intrinsic ratio is defined as

$$Intrinsic\ Ratio = \frac{Simulation\ QT - QT\ in\ Fully\ Coupled\ System}{QT\ in\ ASIA\ System - QT\ in\ Fully\ Coupled\ System}, \qquad (5.17)$$

where the denominator in Eq. (5.17) is the intrinsic gap. In some special cases, if the queueing time can be obtained analytically, we will use the exact queueing time instead of the simulation queueing time in Eq. (5.17).

When the arrival process is Poisson, the intrinsic ratios for all combinations of different arrival rates, first server service times and service time SCVs are shown on the top section of the table in Appendix D, as well as in Figure 5.4.

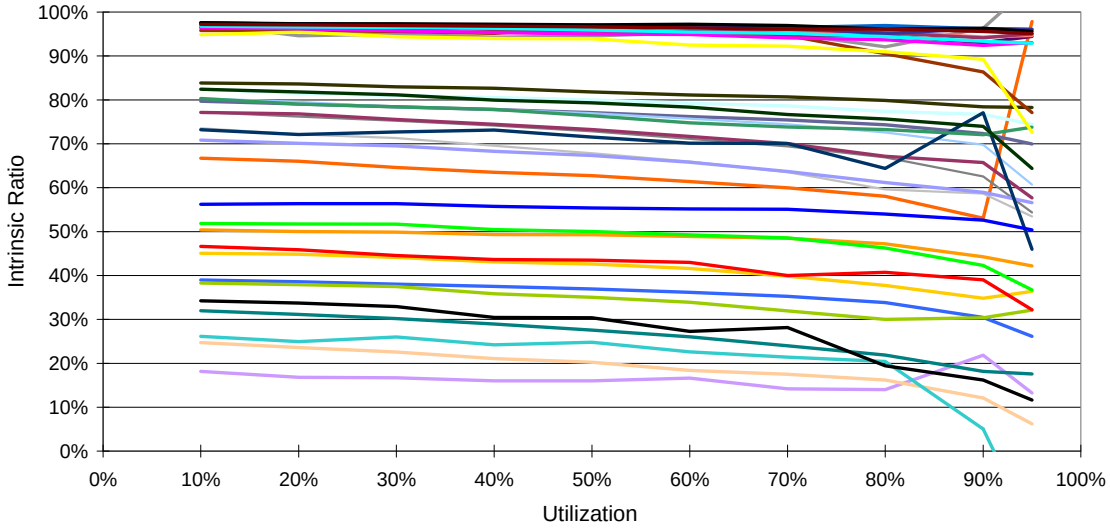

Figure 5.4 Intrinsic ratios vs. utilization for STQB (all cases)

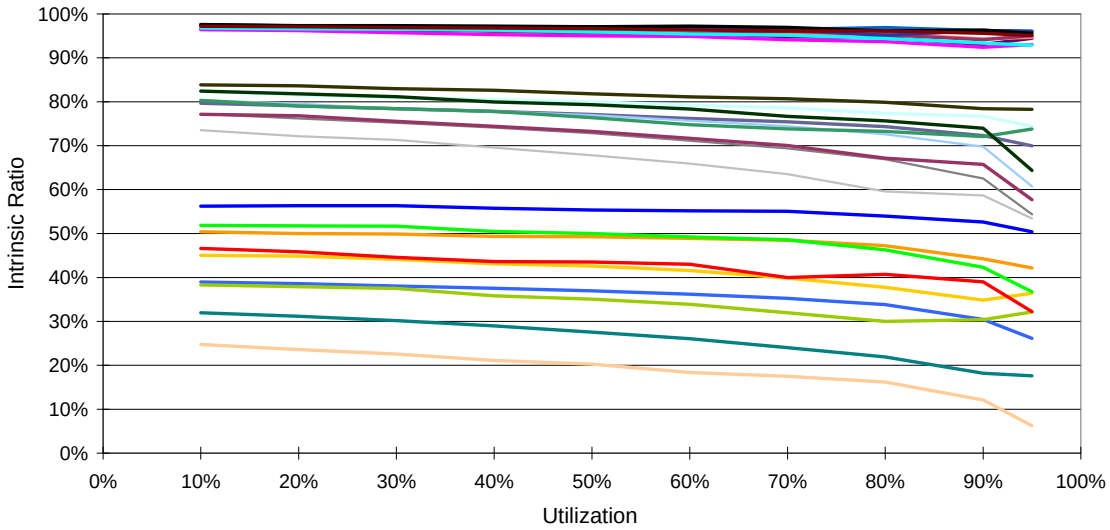

Figure 5.5 Intrinsic ratios vs. utilization for STQB (without 10/30 cases)

In most cases, the intrinsic ratio shows a very regular pattern when utilization is less than 70%. When utilization is high, the intrinsic gap becomes relatively smaller (to the bottleneck queueing time), but the confidence intervals become larger. This explains the irregular trends at high utilization. Furthermore, when the first service time is small (i.e. 10 vs. 30), the intrinsic gap is relatively small compared with the confidence interval,

which may cause an irregular pattern especially at high utilization. It will be interesting to see what Figure 5.4 will become if we remove all the short service time cases (i.e. the first service time is 10 and second service time is 30). The result is shown in Figure 5.5.

Figure 5.5 demonstrates a very regular pattern in general, except in heavy traffic. Indeed, we can even use Figure 5.5 to verify the simulation results: when we see an irregular pattern, we will re-run the simulation by collecting 200 replications with more samples in each replication (e.g. 1,000,000 instead of 200,000) and a longer warm-up period. Our experience shows it always drags the irregular point back into the trend (including all the 10/30 cases). Since this relation is so neat and regular, we call it the "structure of tandem queues". It is important enough to give the following statement:

Observation 5.2 (Nearly-Linear Relationship of Intrinsic Ratio):
The intrinsic ratio is approximately linear across most traffic intensities.

Comparing the intrinsic ratios with the queueing time curve, this pattern is much more linear. In summary, the intrinsic gap and intrinsic ratio give us the following nice properties: (1) While the bottleneck queueing time increases from zero to infinity exponentially as the utilization goes from zero to one, Observation 5.2 tells us its intrinsic ratio possesses a more regular, thus predictable pattern. (2) The relative ratio of the intrinsic gap goes to zero in heavy traffic (i.e. Property 5.1). Furthermore, the intrinsic gap, which is the difference between the ASIA systems and fully coupled systems, can be approximated by Kingman's equation, as long as the initial arrival process is renewal.

Both Property 5.1 and Observation 5.2 give us powerful properties to design new approximate models for the bottleneck queueing time in STQB, as explained in the next section.

5.8.2 Approximate Model for STQB with Small Service Time Variability

Since, from Theorem 5.2, we know the relative ratio of the intrinsic gap goes to zero in heavy traffic, one way to estimate queueing time in heavy traffic is to simply take the average between the upper bound and the lower bound. Obviously, we can do better by taking advantage of Observation 5.2.

If we can predict the intrinsic ratio, we will not only get a better estimate of queueing time in heavy traffic, but also get a better estimate of queueing time for all traffic intensities. Furthermore, we may not need to predict the intrinsic ratio for all traffic intensities. Due to the nearly-linear relationship, knowing any two points is enough to approximate the rest by interpolation or extrapolation.

In general, we need two points to approximate the intrinsic ratios if the slope is steep. However, for Poisson arrivals with service time SCV between 0 and 1, knowing only one point may suffice, since the slope of the intrinsic ratio is close to 0 in all the examined cases. Based on this observation, we can approximate the second station queueing time by

$$QT_2 \cong \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-y_2)\left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \tag{5.18}$$

where y_2 is the intrinsic ratio between the first and second server. When y_2 is 1, Eq. (5.18) is the second station queueing time from the ASIA system. When y_2 is 0, Eq. (5.18) is the second station queueing time from the fully coupled system. Actually, the value of the intrinsic ratio is determined by four factors: initial arrival process, service time ratio (first service time / second service time), and service time SCVs of the first and second servers. If the intrinsic ratio can be approximated by the first and second moment of the initial arrival process and service times, it can be presented as a function of those parameters:

$$\textit{Intrinsic Ratio}: \; y_2 \cong f(\lambda, c_{a1}^2, ST_1/ST_2, c_{s1}^2, c_{s2}^2), \tag{5.19}$$

where ST_i is service time of the i-th server.

Calculating it exactly is difficult, so we will resort to heuristics. Two heuristics are proposed to predict the intrinsic ratio,

1. by the coefficient of variation of the first station service time, or;
2. from the QNA queue time approximation for utilization of 80%.

The first heuristic is motivated by Kingman's approximation and Marshall's equation. From the simulation results in Appendix D, we know the error caused by QNA is around 11%, which is better than QNET. Therefore, it would be interesting to compare Eq. (5.18) with QNA. When the arrival process is Poisson, if Eq. (5.6) is substituted into Eq. (5.1), we obtain

$$QT_2 \cong \left(\frac{c_{a2}^2 + c_{s2}^2}{2}\right)\left(\frac{u_2}{1-u_2}\right)\frac{1}{\mu_2} = \frac{1+c_{s2}^2}{2}\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \frac{\rho_1^2(c_{s1}^2-1)}{2}\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}. \quad (5.20)$$

Comparing Eq. (5.18) and (5.20), we see that only the second terms on the right-hand side are different. We find that y_2 should be related to λ, c_{s1}, and ST_1/ST_2, but not c_{s2} and c_{a1}. By Observation 5.2, we know the intrinsic ratio is insensitive to λ.

To determine if the intrinsic ratio is sensitive to service time ratio, we examine the intrinsic ratio for three cases: (1, 0.1, 0.1), (1, 0.5, 0.5), and (1, 0.9, 0.5), where the values in the bracket mean Poisson arrivals, service time SCV of the first and second servers, respectively. They are marked as group J, G, and K in Figure 5.6, where the numbers, 1, 2, 3, and 4, following the group means the first service times are 10, 20, 25, and 29, respectively. The bottleneck service time is always 30.

In Figure 5.6, we find there are three obvious groups. Within each group, the shorter first service time (e.g. 10) leads to lower intrinsic ratio, while the longer first service time (e.g. 29) leads to higher intrinsic ratio. However, it also shows as long as their SCVs are the same, the different service time ratio does not induce large differences on intrinsic ratio (therefore, they are grouped together). Service time ratio does not play a

dominant role in determining the intrinsic ratio. This can be also observed in the simulation results given in Appendices C and D.

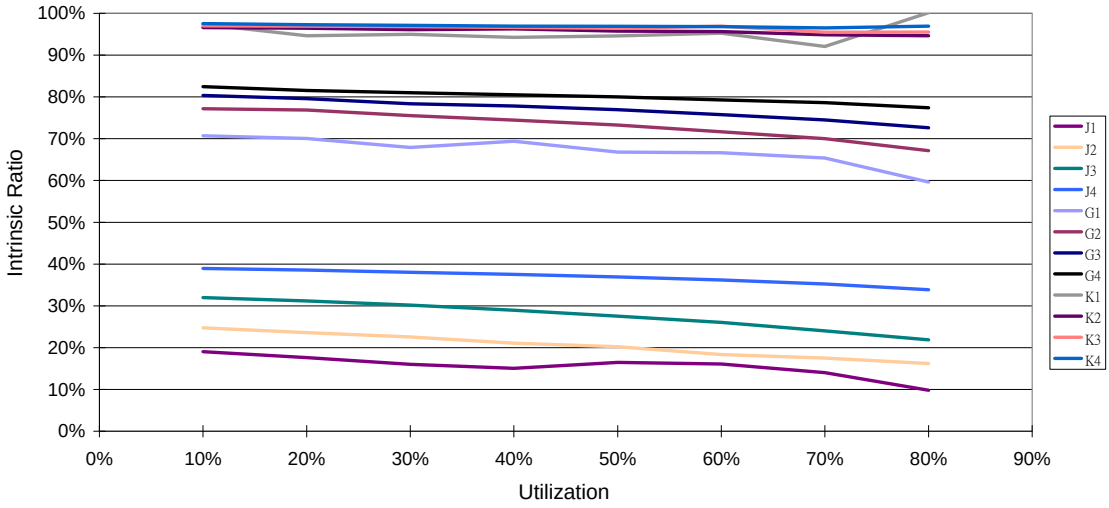

Figure 5.6 Intrinsic ratios vs. utilization (3 special cases)

Therefore, the intrinsic ratio can be approximated simply by c_{s1}. Eq. (5.18) becomes

$$QT_2 \cong \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-c_{s1})\left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \qquad (5.21)$$

Using the simulation results in Appendix D, the average error of Eq. (5.21) is 3.3% (vs. 11.0% for QNA and 13.1% for QNET), and the weighted average error (i.e. approximate error % for the 2nd QT * 2nd QT / System QT) is only 1.8% (vs. 6.3% for QNA and 10.3% for QNET, see more explanation of the weighted average error in Section 5.6). Specifically, at 95% utilization (i.e. in heavy traffic), the error decreases from 12.9% for QNA and 16.0% for QNET to as low as 1.2% for Eq. (5.21). A notable improvement is achieved by using the underlying structure! It is believed the error can be further decreased by improving the quality of intrinsic ratio approximation in Eq. (5.21), such as taking the service time ratio or the second service time SCV into account.

The second heuristic is inspired by Observation 5.1, which says QNA gives the minimum error when traffic intensity is around 70% ~ 80%. By making Eq. (5.1) to be the same as Eq. (5.18), we can obtain the value of y_2 as follows:

$$QT_2 \cong \left(\frac{c_{a2}^2 + c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-y_2)\left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \quad (5.22)$$

where

$$c_{a2}^2 \cong \rho_1^{\,2} c_{s1}^2 + (1-\rho_1^{\,2}) c_{a1}^2.$$

Therefore,

$$y_2 = 1 - \left[\left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - \left(\frac{c_{a2}^2 + c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}\right] \Big/ \left[\left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}\right]. \quad (5.23)$$

Based on Observation 5.1 (in Section 5.6) and Eq. (5.23), we can estimate y_2 when ρ_2 is 80% (more explanation will be given in Section 5.9). By applying them to Eq. (5.22) we can obtain the queueing time approximation by the second heuristic. Compared with the simulation results in Appendix D, the average error of Eq. (5.21) is 5.6%, and the weighted average error is 3.7%. That is not as good as the first heuristic, but still a notable improvement compared with QNA and QNET. The results of these two heuristics demonstrate the power of using the underlying structure.

Another potential approach is to get the intrinsic ratios in heavy and lower traffic first (or obtain a third point at 80% utilization by the second heuristics), then interpolate the rest by those points. Since the intrinsic ratio is much more linear than the queueing time, this approach will give us a much better result than interpolating the queueing time between heavy and light traffic directly. However, calculating intrinsic ratios in heavy and light traffic may not be trivial and thus is left as a direction for future research.

5.8.3 Approximate Model for STQF with Small Service Time Variability

In Section 5.5, we discussed the different impacts of STQB and STQF on system queueing time approximation. While the approximation error of STQB can have a significant impact on system queueing time, the impact from STQF can be negligible. The ratio from Eq. (5.7) of STQF is at least 0.67, and can become large (up to infinity).

$$\frac{E(QT_1)}{E(QT_2)} = \frac{\rho_1/(1-\rho_1)}{\rho_2/(1-\rho_2)} \frac{1/\mu_1}{1/\mu_2} = \frac{\mu_2(\mu_2-\lambda)}{\mu_1(\mu_1-\lambda)}. \tag{5.7}$$

Actually, the ratio of 0.67 comes from one of the two bottleneck cases. If the first server is a distinct bottleneck, the ratio can easily go over 10 or even 100 in heavy traffic (see Appendix E). Since we can get a satisfactory approximation for the first queueing time by Kingman's equation if the initial arrival process is renewal, the approximation error of the second queueing time has less impact on the system queueing time.

In this situation, the results from the parametric-decomposition method may be enough to serve our needs, since large error % contributes a small portion to the system queueing time. However, if we want to adopt the heuristics introduced in Section 5.8.2, the results will not be so appealing. The main reason is that the intrinsic gap does not possess the nice heavy traffic property anymore: Property 5.1 does not apply to STQF.

In both STQB and STQF, the upper bounds are the same, which are the queueing time in their ASIA systems:

$$QT_2(U) = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}. \tag{5.24}$$

However, the lower bound becomes zero, which is the queueing time in its fully coupled system:

$$QT_2(L) = 0. \tag{5.25}$$

Therefore, the intrinsic gap becomes the same as its second queueing time in its ASIA system:

$$IG = QT_2(U) - QT_2(L) = QT_2 = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}. \qquad (5.26)$$

Unfortunately, since the intrinsic gap is associated with the queueing time of the second server (itself), the heavy traffic property does not hold for STQF.

As in Appendix D, we conduct 9 experiments for STQF. The first service time is always 30, and the second service times are 10, 20, 25, and 30. The service time SCVs equally spread out between 0 and 1 (0.1 for Low, 0.5 for Medium, and 0.9 for High levels) and the arrival process is Poisson. The other simulation settings are the same as previous cases. The results are shown in Appendix E. In all 9 cases, the largest half-width 90% confidence interval of queueing time is 1.80% (from (1, 0.1, 0.9) & 30/30 at 95% utilization), and most of them are smaller than 0.5%. The intrinsic ratios for the cases in Appendix E are shown in Figure 5.7.

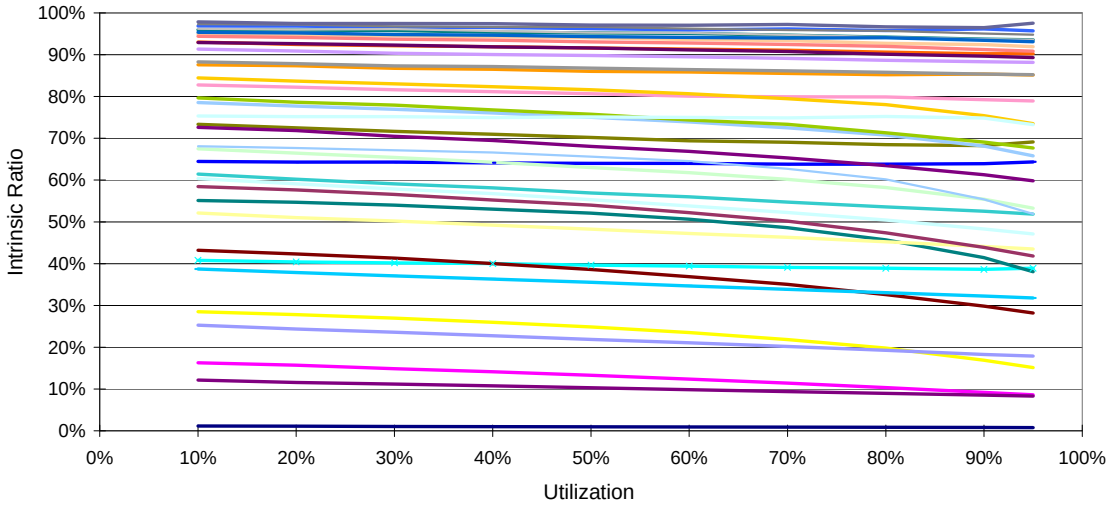

Figure 5.7 Intrinsic ratios vs. utilization for STQF (Appendix E)

Although the heavy traffic property does not apply to STQF, the nearly-linear relationship of the intrinsic ratio still holds as shown in Figure 5.7. For STQF, the quality

of approximations depends on how accurately we can estimate the intrinsic ratio. Therefore,

$$QT_2 = x_2\left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}, \tag{5.27}$$

where *Intrinsic Ratio* is $x_2 \cong f(\lambda, c_{a1}^2, ST_1/ST_2, c_{s1}^2, c_{s2}^2)$.

By directly observing the simulation results in Appendix E, we find the intrinsic ratio is very sensitive to the service time (*ST*) ratio (i.e. ST_1/ST_2). Since the service time ratio plays an important role in determining the intrinsic ratio of STQF, we cannot ignore it anymore as we did in STQB. Thus, the first heuristic in STQB does not work well for STQF in general.

In order to apply the second heuristic, we need first to investigate the errors caused by QNA in STQF. From Appendix E, when there are two bottlenecks (i.e. 30/30), QNA works well when utilization is around 80%. However, when there is a distinct bottleneck and the second service time is relatively shorter than the first service time, QNA tends to induce positive errors (i.e. over-estimate the true value). Furthermore, the errors become smaller when utilization approaches 1.

Observation 5.3:

When the initial arrival process is Poisson and service time SCV is smaller than 1, the parametric-decomposition approximation tends to induce positive errors, and the error decreases when the utilization approaches 1 in simple tandem queues with distinct front-end bottlenecks.

Similar to the second heuristic in STQB, we can approximate the intrinsic ratio (x_2) by QNA as follows.

$$QT_2 \cong \left(\frac{c_{a2}^2 + c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} = x_2\left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}, \tag{5.28}$$

where

$$c_{a2}^2 \cong \rho_1^2 c_{s1}^2 + (1-\rho_1^2) c_{a1}^2.$$

Therefore,

$$x_2 = \left(\frac{c_{a2}^2 + c_{s2}^2}{2} \right) / \left(\frac{1 + c_{s2}^2}{2} \right). \tag{5.29}$$

Based on Observation 5.3, when $ST_1 = ST_2$, we approximate x_2 by assigning 0.8 to ρ_2 (which is the same as STQB). When there is a distinct bottleneck in STQF, we simply approximate x_2 by assigning 0.99 to ρ_2. By substituting for x_2 to Eq. (5.27), we can approximate the queueing times at other traffic intensities. Based on the simulation results in Appendix E, the average error for this approach is 74.4%, but the weighted average error is only 1.6%. Although the average error is large, the weighted average is small anyway, because the first queueing time is relatively longer than the second queueing time in STQF.

The average approximation error from QNA is 261.9% and the weighted average error is 6.4%. Most of the error comes from the (1, 0.1, 0.1) and 30/10 case. The average approximation error from QNET is 80.1% and the weighted average error is 1.2%. QNA performs the worst among the three. The performances of QNET and the intrinsic ratio (IR) approach are about the same. Nevertheless, due to the small weighted average errors, the approximate errors from STQF have very little impact on system queueing time.

5.8.4 Approximate Model for STQB with Large Service Time Variability

In all previous examples, we focused on the cases with Poisson arrivals and service time SCV smaller than 1. In this section, we want to discuss the case with Poisson arrivals and service time SCV greater than 1.

In this situation, based on Conjecture 5.2b, the ASIA system performs like a lower bound instead of an upper bound. While the fully coupled system still gives a

conservative lower bound, the ASIA system gives us a higher one in this specific situation. The intrinsic ratio is still defined as

$$Intrinsic\ Ratio = \frac{Simulation\ QT - QT\ in\ Fully\ Coupled\ System}{QT\ in\ ASIA\ System - QT\ in\ Fully\ Coupled\ System}, \qquad (5.17)$$

Since the intrinsic gap is the difference between its ASIA system and fully coupled system, the heavy traffic property still holds for STQB, but the intrinsic ratio would be greater than 1 instead of between 0 and 1.

As before, we conduct nine experiments in this case. The second service time is always 30, and the first service times can be 10, 20, 25, or 29. The service time SCV is either 2, 5 or 8 (Low, Medium, and High), and the arrival process is Poisson. In Appendix F, each observation is the average of 100 replications. Each replication is the average of 200,000 ~ 1,000,000 data points after a warm up period of 50 years (87,600 ~ 832,200 data points) or longer. The largest half-width 90% confidence interval of queueing time is 3.04% (from (1, 2, 8) & 25/30 at 95% utilization), and most are smaller than 1% except for some high utilization cases.

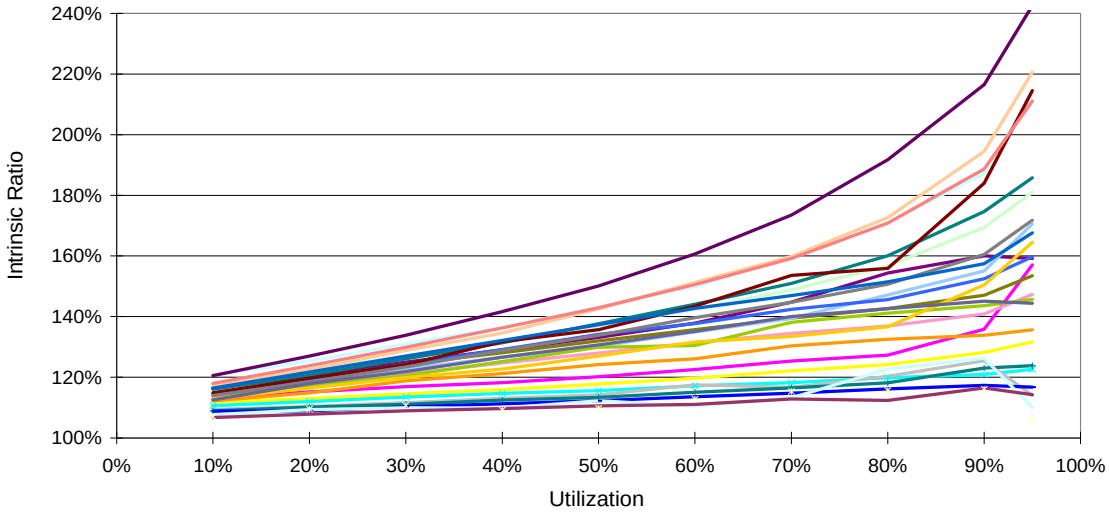

Figure 5.8 Intrinsic ratios vs. utilization with large service time variability (Appendix F)

If we filter out the cases where the first service time is 10, the intrinsic ratios for the cases in Appendix F are shown in Figure 5.8. Although the nearly-linear relationship of Observation 5.2 still holds for the intrinsic ratio, its slope is not close to zero. We need at least two points in order to interpolate or extrapolate the other intrinsic ratios. However, because QNA can only give a reliable approximation at one traffic intensity, we can only adopt the single point approach here. This issue will be solved in Chapter 6, where the intrinsic ratios are estimated by historical queueing times.

By observing the simulation results in Appendix F, QNA performs well around 70 ~ 80% for STQB even when the service time SCV is greater than 1. Specifically, if the non-bottleneck service time (i.e. 10 and 20) is relatively shorter than the bottleneck service time (i.e. 30), QNA performs well around 70%. When the non-bottleneck service time (i.e. 25 and 29) is close to the bottleneck service time, QNA performs well around 80%. More discussion will be given in Section 5.9.

Observation 5.4:

When the initial arrival process is Poisson and service time SCV is greater than 1, Parametric-decomposition approximation performs well when the bottleneck traffic intensity is around 70% ~ 80% in simple tandem queues with backend bottlenecks.

To simplify the algorithm, we calculate y_2 only when ρ_2 is 80% by Eq. (5.23). However, we should keep in mind that the results may be improved if we calculate y_2 at 70% utilization when the non-bottleneck service time is relatively short.

Since the utilizations of 80% are pretty high, if we are only interested in the performance in heavy traffic, the second heuristic with single point approach should be adequate with the help of the heavy traffic property of the intrinsic gap (Property 5.1). In Appendix F, the average error percentage of the second heuristic in heavy traffic is 1.8%. The overall average approximation error is 14.6% and the weighted error is 9.4%.

The average approximation error from QNA is 8.3% and the weighted error is 5.6%. The average approximation error from QNET is 10.3% and the weighted error is 6.9%. The performance of QNA is better in terms of the average errors. However, if we look at their performance in heavy traffic, the second heuristic (1.8%) is better then QNET (2.7%) and QNA (21.3%). Therefore, a potential approach is to use QNA when utilization is less than 80% and to use the second heuristic in heavy traffic.

5.9 Structure of Errors in Kingman's Approximation

In this section, we want to elaborate further on Observations 5.1, 5.3 and 5.4. From Appendix D, we can observe that QNA performs well around 70% ~ 80% for STQB when service time SCV is smaller than 1 (Observation 5.1). From Appendix E, we observe that, in STQF, if the service time SCV is smaller than 1, the errors of QNA approximations decrease when the utilization approaches 1 (Observation 5.3).

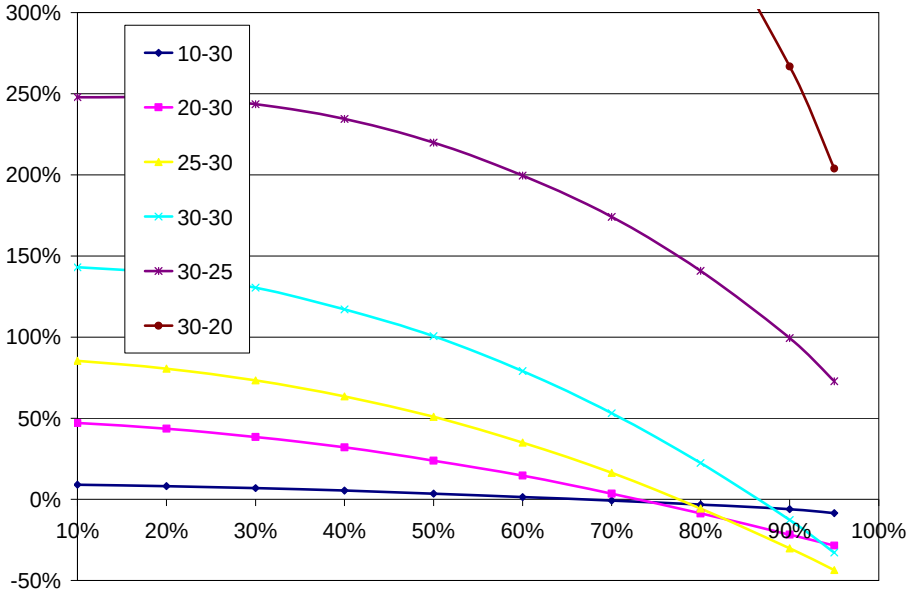

Figure 5.9 Errors of Kingman's approximation at the second server for SCV(0.1, 0.1)

Figure 5.9 shows the approximation errors of the second queueing time in simple tandem queues caused by QNA when both service time SCVs are 0.1, represented as SCV(0.1, 0.1), where the first 0.1 in the bracket is the service time SCV of the first server and the second 0.1 is the service time SCV of the second server in a simple tandem queue. The initial arrival process is Poisson. The x-axis is utilization (from 10% to 100%) and the y-axis is the percentage errors. The bottleneck service time is always 30. Each represents the results of different service time combinations: the left is the service time of the first server, and the right is the service time of the second server, i.e. ST_1-ST_2.

The lowest three lines are the errors of STQB, which all give negligible error between 70 ~ 80% utilization. The three upper lines are the errors of STQF. Except for the 30-30 case, in which 0 error occurs at ~87%, all the rest have positive errors across all utilizations, but with the minimum error when utilization approaches 1. The errors for the 30-10 case are very large (over 3000%) and are not shown.

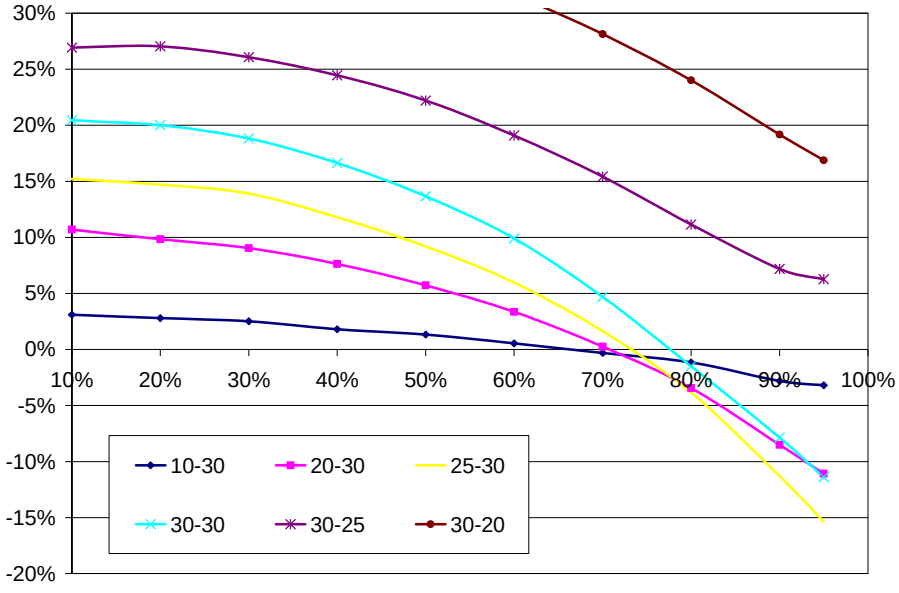

Figure 5.10 Errors of Kingman's approximation at the second server for SCV(0.5, 0.5)

Figure 5.10 demonstrates the errors for SCV(0.5, 0.5). Except for the service time SCV, all the other conditions are the same as the previous case. The lowest three lines are the errors of STQB, in which negligible errors occur between 65 ~ 75% utilization. The upper three lines are the errors of STQF. Except for the 30-30 case, in which negligible errors occur at ~78%, all the rest have positive errors across all utilizations, but with the minimum error when utilization approaches 1. The errors for the 30-10 case are very large (over 60%) and are not shown in the figure. An important observation is that although the intersections at the x-axis look similar, the scale of the y-axis is very different (30% vs. 300%). The conclusion from these two charts is that although (0.1, 0.1) gives larger errors, it still gives accurate results around 70 ~ 80% for STQB as occurred in the case of (0.5, 0.5). Furthermore, they both induce positive errors for STQF, which is consistent with Observation 5.3.

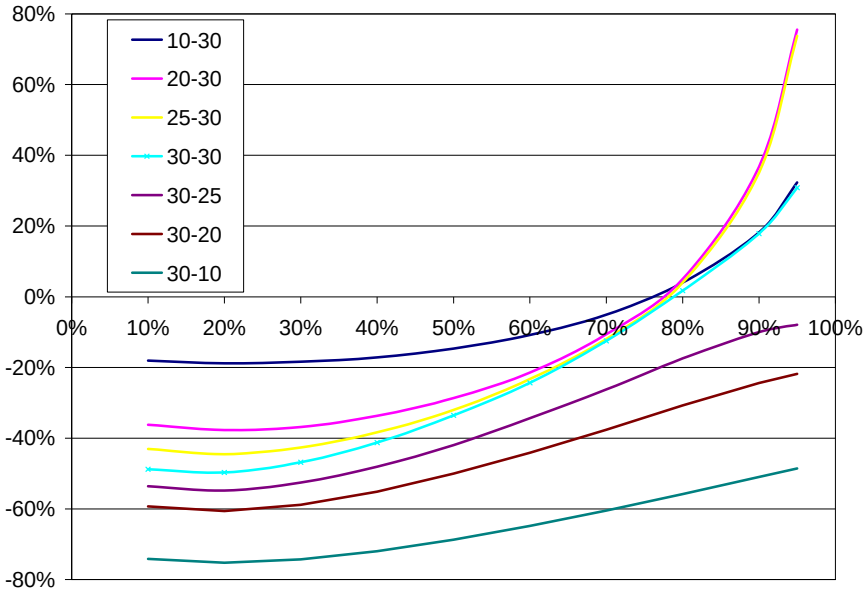

Figure 5.11 Errors of Kingman's approximation at the second server for SCV(8, 0.5)

The above cases give us more solid evidence to support Observations 5.1 and 5.3. Similar analysis can be done for the cases with service time SCV larger than 1. Figure 5.11 demonstrates the errors for SCV(8, 0.5). The top three lines at 10% utilization are the errors of STQB, in which negligible errors occur between 75 ~ 80% utilization. The three lower lines are the errors of STQF. Except for the 30-30 case, in which negligible errors occur at ~78%, all the rest have negative errors across all utilizations, but with the minimum error when utilization approaches 1. The errors in Figure 5.11 seem symmetric to Figure 5.10 and cross the x-axis at around 70 ~ 80% utilization.

The last example is SCV(8, 2) in Figure 5.12. The top three lines at 10% utilization are the errors of STQB, in which 0 error occurs between 68 ~ 75% utilization. Figure 5.11 and 5.13 give more solid demonstration of Observation 5.4. From the above analysis, it seems a better approximation of intrinsic ratios can be achieved by analyzing the structure of errors in a more detailed manner. It is left as a direction for future research.

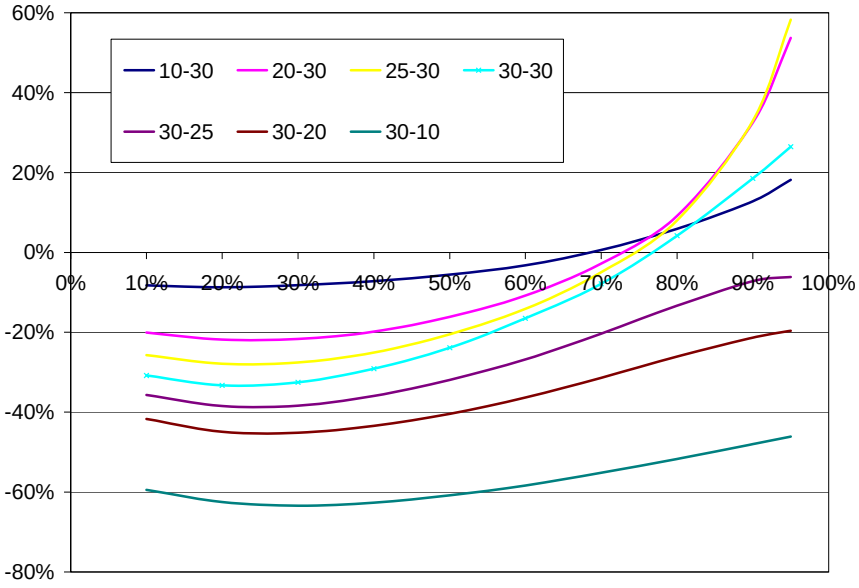

Figure 5.12 Errors of Kingman's approximation at the second server for SCV(8, 2)

5.10 Conclusions

In this chapter, we have proposed new approximate models for simple tandem queues by using the nice properties from the intrinsic gap and intrinsic ratio. The intrinsic gap (with the heavy traffic property in STQB) acts as a transformer, converting an exponential queueing time curve into a nearly-linear intrinsic ratio. The model performs very well for the simple tandem queues with backend bottlenecks, which is about the most important case from a practical perspective. The intrinsic ratio based approximate models perform well in most examined categories of test problems. QNA and QNET also performed well in some categories, but poorly in others. Comparing QNA with QNET in our test problems, neither clearly dominates the other. A similar observation will be made later in Chapter 6, where we study the behavior of multiple single-server queues in series.

Due to the dependence among workstations, queueing time is difficult to analyze exactly in practical manufacturing systems. In approximating system queueing times, there can be direct and indirect approaches. A direct approach to handling the dependence is based on the intrinsic ratio. The indirect approach is to treat the dependence by making additional assumptions, such as product-form solutions or Brownian motion. Under FIFO dispatching scheme, when service time and inter-arrival time are exponential, all servers behave independently. Likewise, when all servers work in heavy traffic, the first and second moments of service time and inter-arrival time are enough to describe system behaviors exactly. However, in practical manufacturing systems, these cases seldom occur; departure processes are not renewal, and the first two moments are not enough to describe system behavior.

The additional assumptions of the indirect approach allow the queueing networks to be analyzed exactly. The errors of the indirect approaches depend on the impact from those assumptions, i.e. how much the analyzed system deviates from the modeled system.

Without making indirect assumptions, this chapter presents an alternative approach based on the concept of the intrinsic ratio, and thus attempts to deal directly with the dependence among workstations. We have demonstrated a way to deal with dependence by taking advantage of the observed underlying structures, and have achieved notable improvement in the approximate errors. However, more work is needed. We want to gain more understanding of the intrinsic ratio in order to calculate it more accurately or even exactly. Furthermore, the behavior of intrinsic ratio in other situations, such as multiple products or multiple-server queues has not been investigated. Those topics are left for future research.

In the next chapter, we will extend the results of simple tandem queues to production lines.

CHAPTER 6

BEHAVIOR OF MULTIPLE SINGLE-SERVER QUEUES IN SERIES

The final test of a theory is its capacity to solve the problems which originated it.

~ Dantzig

6.1 Introduction

A production line consists of multiple workstations in series, where each workstation may consist of a group of identical machines. Based on Papadopoulos, Heavey and Browne (1993), production lines are systems with asynchronous part transfer, in contrast to transfer lines, which have synchronous part transfer.

From the viewpoint of system complexity, a production line is on an intermediate level between general queueing networks and simple tandem queues. Multiple single-server queues in series form a special case of production lines, where each workstation is composed of only one single server. In this chapter, we expand the results of simple tandem queues in the previous chapter to multiple queues in series. Three new approximate models are developed, according to the underlying structure of tandem queues. The new models are compared with other results in the literature.

As reviewed in Chapter 4, several approximate models have been developed to analyze the performance of queueing networks. Queueing Network Analyzer (QNA), a parametric-decomposition approach, proposed by Whitt (1983) is based on assuming the non-renewal departure process is renewal. Motivated by heavy traffic theory, Harrison and Nguyen (1990) proposed the QNET method to approximate queueing networks. Dai and Harrison (1992) developed the QNET algorithm further to obtain numerical results. However, the computational complexity of the QNET algorithm grows in the size of the network. In order to overcome these computational issues, Dai, Nguyen and Reiman

(1994) developed the Sequential Bottleneck Decomposition (SBD) method to reduce the computational complexity by grouping workstations with similar utilizations, and limiting these sub-networks to a reasonable size.

In the present study we take a different approach. We propose approximate models for the queueing time at each server, and an aggregate model to approximate total system queueing time. Based on this model, a way to analyze the dependence among servers in general tandem queues is introduced.

The structure of this chapter is as follows: In Section 6.2, the underlying structure of many single server queues in series is introduced. The corresponding approximate models are developed in Section 6.3. A procedure to implement the approximate models is introduced in Section 6.4. Performance of the approximate models is compared with previous approaches in Section 6.5. The dependence among single server queues in series is discussed in Section 6.6.

6.2 Structure of Multiple Single-Server Queues in Series

In Chapter 5, we proposed a new model to approximate the queueing time of simple tandem queues by taking advantage of their underlying structure, in particular Observation 5.2, the nearly-linear relationship. It is important to know if the observed structure carries over to tandem queues with more than two servers.

In order to know if this underlying structure still exists for production lines, we conduct simulations on a production line with five single-server queues in series. The mean service times are 20, 25, 30, 25 and 20 with Erlang(2) distribution (i.e. service time SCV is 0.5). The arrival process is Poisson. Traffic intensity of the bottleneck (i.e. the third server) varies from 10% up to 95%. 100 replications are conducted at each specific input rate. Each replication is composed of 200,000 data points after a 50 year warm-up period (i.e. 87,600 – 832,200 data points are discarded depending on input rates). For

different utilizations, the mean queueing times of each server and their half width 90% confidence intervals are shown in Table 6.1.

Table 6.1 Mean queueing times and the 90% confidence intervals

Util 3	QT 1	90% CI	QT 2	90% CI	QT 3	90% CI	QT 4	90% CI	QT 5	90% CI
10%	1.07	0.22%	1.49	0.21%	2.11	0.17%	1.27	0.19%	0.73	0.23%
20%	2.31	0.18%	3.27	0.19%	4.74	0.15%	2.76	0.17%	1.56	0.18%
30%	3.75	0.14%	5.45	0.16%	8.11	0.15%	4.55	0.16%	2.48	0.17%
40%	5.46	0.13%	8.16	0.16%	12.59	0.17%	6.75	0.16%	3.57	0.17%
50%	7.51	0.14%	11.66	0.17%	18.92	0.16%	9.54	0.18%	4.84	0.17%
60%	10.01	0.17%	16.37	0.18%	28.57	0.22%	13.23	0.17%	6.37	0.16%
70%	13.14	0.15%	22.96	0.17%	44.90	0.27%	18.30	0.21%	8.25	0.16%
80%	17.17	0.20%	33.06	0.25%	78.57	0.36%	25.72	0.22%	10.62	0.18%
90%	22.50	0.20%	49.89	0.34%	183.81	0.85%	37.60	0.24%	13.70	0.18%
95%	25.86	0.20%	63.91	0.30%	398.97	1.43%	46.59	0.31%	15.61	0.21%

Based on the reduction methods and the definition of ASIA systems in Chapter 5, when the service time SCVs are smaller than 1 and the arrival process is Poisson, the upper bounds and lower bounds on queueing time for each server can be calculated as follows:

$$QT_2^U = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}, \tag{6.1}$$

$$QT_2^L = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - \left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \tag{6.2}$$

$$QT_3^U = \left(\frac{1+c_{s3}^2}{2}\right)\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3}, \tag{6.3}$$

$$QT_3^L = \left(\frac{1+c_{s3}^2}{2}\right)\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} - \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - \left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \tag{6.4}$$

$$QT_4^U = \left(\frac{1+c_{s4}^2}{2}\right)\left(\frac{\rho_4}{1-\rho_4}\right)\frac{1}{\mu_4}, \tag{6.5}$$

$$QT_4^L = 0, \tag{6.6}$$

$$QT_5^U = \left(\frac{1+c_{s5}^2}{2}\right)\left(\frac{\rho_5}{1-\rho_5}\right)\frac{1}{\mu_5}, \tag{6.7}$$

$$QT_5^L = 0. \tag{6.8}$$

Because the third server is the bottleneck, the lower bounds of queueing time for the forth and fifth server are zero. The upper bounds, lower bounds, intrinsic gaps and intrinsic ratios of each server are shown in Table 6.2. An interesting observation is that the computed lower bound can be negative (for the third server) when utilization is low.

Table 6.2 Lower bounds, upper bounds, intrinsic gaps and intrinsic ratios

	2nd Server				3rd Server				4th Server				5th Server			
Util 3	LB	UB	Gap	IR	LB	UB	Gap	IR	LB	UB	Gap	IR	LB	UB	Gap	IR
10%	0.63	1.70	1.07	79.9%	-0.06	2.50	2.56	84.8%	0.00	1.70	1.70	74.5%	0.00	1.07	1.07	68.2%
20%	1.44	3.75	2.31	79.3%	0.05	5.63	5.58	84.1%	0.00	3.75	3.75	73.7%	0.00	2.31	2.31	67.4%
30%	2.50	6.25	3.75	78.6%	0.45	9.64	9.20	83.3%	0.00	6.25	6.25	72.7%	0.00	3.75	3.75	66.2%
40%	3.92	9.38	5.45	77.8%	1.38	15.00	13.62	82.3%	0.00	9.38	9.38	72.0%	0.00	5.45	5.45	65.4%
50%	5.89	13.39	7.50	76.9%	3.33	22.50	19.17	81.3%	0.00	13.39	13.39	71.2%	0.00	7.50	7.50	64.5%
60%	8.75	18.75	10.00	76.2%	7.38	33.75	26.37	80.4%	0.00	18.75	18.75	70.6%	0.00	10.00	10.00	63.7%
70%	13.13	26.25	13.13	74.9%	16.41	52.50	36.09	79.0%	0.00	26.25	26.25	69.7%	0.00	13.13	13.13	62.9%
80%	20.36	37.50	17.14	74.1%	39.78	90.00	50.22	77.2%	0.00	37.50	37.50	68.6%	0.00	17.14	17.14	62.0%
90%	33.75	56.25	22.50	71.7%	130.11	202.50	72.39	74.2%	0.00	56.25	56.25	66.8%	0.00	22.50	22.50	60.9%
95%	45.34	71.25	25.91	71.7%	337.73	427.50	89.77	68.2%	0.00	71.25	71.25	65.4%	0.00	25.91	25.91	60.3%

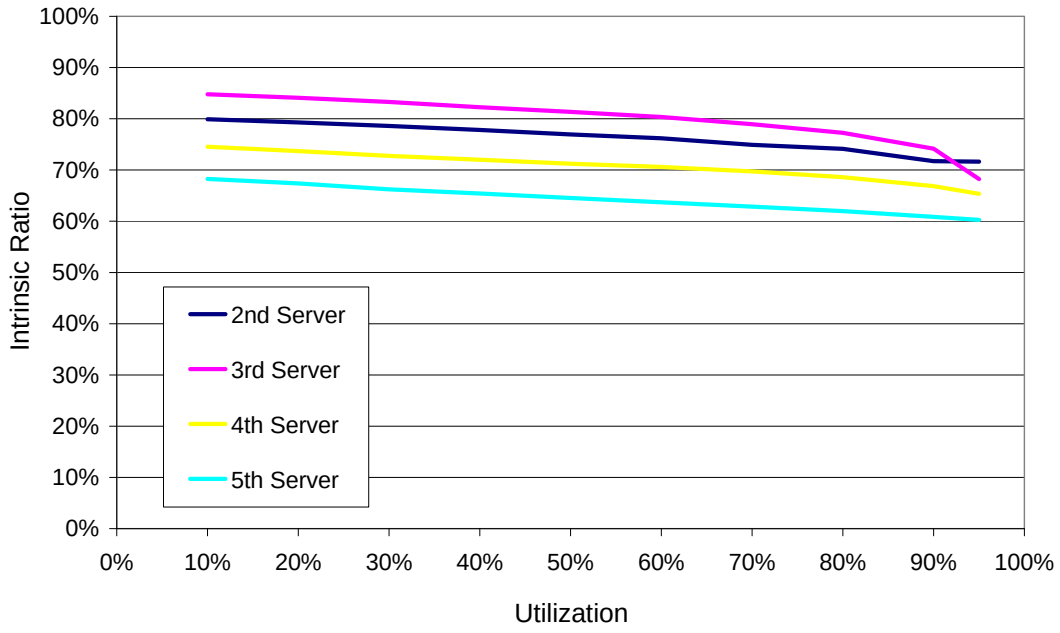

Figure 6.1 Intrinsic ratios of five single queues in series

Figure 6.1 shows the intrinsic ratios of each server at different utilizations. An important finding is that although only the initial arrival process is renewal, the nearly-linear relationship is preserved for the five queues in series. This indicates that Observation 5.2 might be applicable to multiple single queues in series. This conjecture will be tested more extensively in Section 6.5.

6.3 Approximate Models for Multiple Single-Server Queues in Series

Similar to the analysis in Section 5.8, in this section, we are going to derive approximate models for production lines. We begin by looking at three different cases of three single queues in series with *Poisson arrivals*: (1) $\mu_1 > \mu_2 > \mu_3$ (2) $\mu_2 >= \mu_1 > \mu_3$ (3) $\mu_3 >= \mu_2 >= \mu_1$, where μ_i is the service rate of the *i*-th server as shown in Figure 6.2.

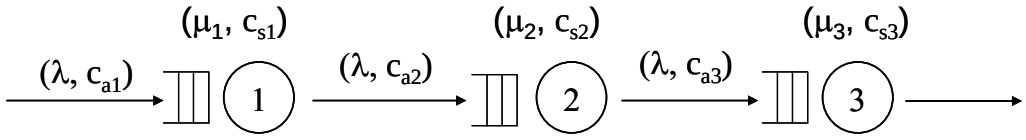

Figure 6.2 Three single server queues in series

Case 1: $\mu_1 > \mu_2 > \mu_3$

In Chapter 5, we have proposed the approximate models for simple tandem queues. For three single queues in series, the formulae of the first two servers are the same as the ones in simple tandem queues. The queueing times of the first server can be calculated by the P-K formula:

$$QT_1 = \left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} = \alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \qquad (6.9)$$

where $\alpha_i = (c_{a1}^2 + c_{si}^2)/2$ for the *i*-th server. Since $\mu_2 < \mu_1$ (i.e. STQB), the queueing time of the second server can be expressed as

$$QT_2 = \left(\frac{1+c_{s2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-y_2)\left(\frac{1+c_{s1}^2}{2}\right)\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}, \qquad (6.10)$$

where $y_2 \cong f(\lambda, c_{a1}^2, ST_1/ST_2, c_{s1}^2, c_{s2}^2)$ from Section 5.8.2.

Eq. (6.10) can be also expressed as

$$QT_2 = \alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-y_2)\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}. \qquad (6.11)$$

If we approximate y_2 by a constant, $\overline{y_2}$, Eq. (6.11) becomes

$$QT_2 \cong \alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-\overline{y_2})\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}. \qquad (6.12)$$

To approximate the queueing time of the third server, we first calculate the queueing time of its associated ASIA system,

$$QT_3(U) = \left(\frac{1+c_{e3}^2}{2}\right)\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} = \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3}. \qquad (6.13)$$

Since $\mu_1 > \mu_2 > \mu_3$, the queueing time of its fully coupled system is

$$\begin{aligned} QT_3(L) &= QT_3 - (QT_1 + QT_2) \\ &= \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} - \left[\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + \left\{\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-y_2)\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}\right\}\right] \qquad (6.14) \\ &= \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} - \left[\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + y_2\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}\right]. \end{aligned}$$

Using Observation 5.2, the third queueing time can be approximated as

$$QT_3 = \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} - (1-y_3)\left[\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + y_2\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}\right]. \qquad (6.15)$$

The system queueing time is the summation of Eq. (6.9), (6.11) and (6.15),

$$\sum_{i=1}^{3} QT_i = y_2 y_3 \alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + y_3\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3}. \qquad (6.16)$$

Case 2: $\mu_2 >= \mu_1 > \mu_3$

The queueing times of the first servers can be determined by Eq. (6.9). Since $\mu_2 >= \mu_1$ (i.e. STQF), similar to Eq. (5.23), the queueing time of the second server can be expressed as

$$QT_2 = x_2\left(\frac{1+c_{e2}^2}{2}\right)\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} = x_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}, \tag{6.17}$$

where $x_2 \cong f(\lambda, c_{a1}^2, ST_1/ST_2, c_{e1}^2, c_{e2}^2)$ from Section 5.8.3.

To approximate the queueing time of the third server, we first calculate the queueing time of its ASIA system,

$$QT_3(U) = \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3}. \tag{6.18}$$

Since $\mu_2 >= \mu_1 > \mu_3$, the queueing time of its fully coupled system is

$$QT_3(L) = \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} - \left[\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + x_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}\right]. \tag{6.19}$$

Due to Observation 5.2, the third queueing time can be approximated as

$$QT_3 = \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3} - (1-y_3)\left[\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + x_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}\right]. \tag{6.20}$$

The system queueing time is the summation of Eq. (6.9), (6.17) and (6.20),

$$\sum_{i=1}^{3} QT_i = y_3\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + x_2y_3\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \alpha_3\left(\frac{\rho_3}{1-\rho_3}\right)\frac{1}{\mu_3}. \tag{6.21}$$

Case 3: $\mu_3 >= \mu_2 >= \mu_1$

The queueing times of the first servers is determined by Eq. (6.9). Since $\mu_2 >= \mu_1$ (i.e. STQF), the queueing time of the second server is determined by Eq. (6.17). To approximate the queueing time of the third server, we first calculate the queueing time of its ASIA system,

$$QT_3(U) = \alpha_3 \left(\frac{\rho_3}{1-\rho_3} \right) \frac{1}{\mu_3}. \quad (6.22)$$

Since $\mu_3 >= \mu_2 >= \mu_1$, the queueing time of its fully coupled system is

$$QT_3(L) = 0. \quad (6.23)$$

Using Observation 5.2, the third queueing time can be approximated as

$$QT_3 = x_3 \alpha_3 \left(\frac{\rho_3}{1-\rho_3} \right) \frac{1}{\mu_3}. \quad (6.24)$$

The system queueing time is the summation of Eq. (6.9), (6.17) and (6.24),

$$\sum_{i=1}^{3} QT_i = \alpha_1 \left(\frac{\rho_1}{1-\rho_1} \right) \frac{1}{\mu_1} + x_2 \alpha_2 \left(\frac{\rho_2}{1-\rho_2} \right) \frac{1}{\mu_2} + x_3 \alpha_3 \left(\frac{\rho_3}{1-\rho_3} \right) \frac{1}{\mu_3}. \quad (6.25)$$

6.3.1 Approximate Model for Each Server in Tandem Queues

In a fully coupled system, all servers behind (or after) the system bottleneck are classified as non-bottlenecks and have zero queueing times. If a server is identified as the next bottleneck before (or in front of) the system bottleneck, all servers between the main and the second bottleneck are classified as non-bottlenecks and have zero queueing times.

Based on the above insight and the observation from the previous three cases, we have Procedure 6.1 to approximate the queueing times of n servers in series.

Due to the property of fully coupled systems, the above procedure identifies the next bottlenecks within each subsystem, where a subsystem is composed of the servers from the first server to the newest identified bottleneck (not included). At the beginning, when no bottleneck has been identified, the subsystem is the same as the original system. The subsystem then gradually becomes smaller until the subsystem is solely composed of one single server, which is the first server of the tandem queue.

Procedure 6.1 (Queueing Time of Each Server in Tandem Queues):

Stage I: Decomposition by bottlenecks	Explanation
1. Identify the index of the system bottleneck server (BN_1), where μ_{BN_1} = min μ_i, for i = 1 to n. Let k = 1. If there is more than one BN_1 (with the same μ), mark the one with the smallest sequence number.	Find the 1st bottleneck. All the servers after the system bottleneck are non-bottlenecks. When there are two identical bottlenecks, since the later one has zero queueing time in the fully coupled system, mark the later one as a non-bottleneck.
2. Identify the index of the next bottleneck server (BN_{k+1}) in front of the previous one (BN_k), where $\mu_{BN_{k+1}}$ = min μ_i, for i = 1 to $BN_k - 1$. If there is more than one BN_{k+1} (with the same μ), mark the one with the smallest sequence number.	Find the next bottleneck in front of the previously identified bottleneck. All the servers between the two bottlenecks are non-bottlenecks.
3. If BN_{k+1} = 1, stop. Otherwise, let $k = k + 1$, go to 2.	Stop when the 1st server is assigned to be a bottleneck (the p-th bottleneck in Figure 6.3).
Stage II: Determining the formula to use	
4. Let $QT_1 = \alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1}$, and i = 2.	Calculate the queueing time for the first server.
5. If server i is marked as a bottleneck, then $QT_i = \alpha_i\left(\frac{\rho_i}{1-\rho_i}\right)\frac{1}{\mu_i} - (1-y_i)\sum_{j=1}^{i-1} QT_j$. Otherwise, $QT_i = x_i\alpha_i\left(\frac{\rho_i}{1-\rho_i}\right)\frac{1}{\mu_i}$.	Calculate the queueing times for the other servers. If the server is identified as a bottleneck, use the model similar to STQB. Otherwise, use the model of STQF.
6. If $i = n$, stop. Otherwise, let $i = i + 1$, go to 5.	

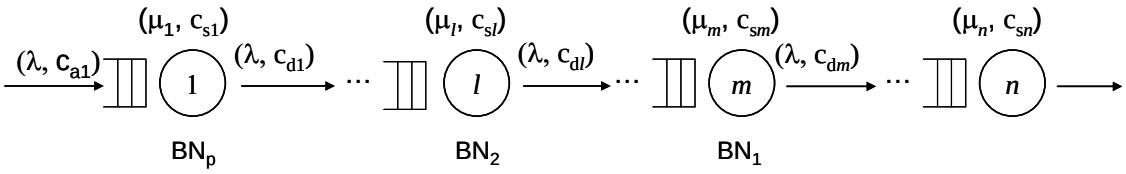

Figure 6.3 N single queues in series

From Procedure 6.1, we can also see the importance of the ASIA systems. The queueing time of each server is determined by the intrinsic ratios and its variability in the ASIA systems, but not by the true variability in the original systems.

6.3.2 Approximate Model for System Queueing Time

By summing up the queueing times of all servers, the system queueing time can be approximate by the following formula:

$$\sum_{i=1}^{n} QT_i = f_1\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + f_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \ldots + f_n\alpha_n\left(\frac{\rho_n}{1-\rho_n}\right)\frac{1}{\mu_n}, \qquad (6.26)$$

where f_i is called a contribution factor, since it represents the percentage of server i's ASIA system queueing time contributing to the overall system queueing time.

Definition 6.1 (Contribution Factor of the ASIA system queueing time):
The contribution factor (f_i) describes the portion of the contribution from each server's ASIA system queueing time to the overall system queueing time.

f_i can be determined based on the following procedure.

Procedure 6.2 (System Queueing Time of Queues in Series):
Stage I: Decomposition by bottlenecks

1. Identify the system bottleneck (BN_1), where

   μ_{BN_1} = min μ_i, for i = 1 to n. Let k = 1.

   - If there is more than one BN_1 (with the same μ), mark the one with the smallest sequence number.
2. Identify the next bottleneck (i.e. BN_{k+1}) in front of the previous one (i.e. BN_k), where $\mu_{BN_{k+1}}$ = min μ_i, for i = 1 to $BN_k - 1$.

   - If there is more than one BN_{k+1} (with the same μ), mark the one with the smallest sequence number.

3. If $BN_{k+1} = 1$, stop. Otherwise, let $k = k + 1$, go to 2.

Stage II: Determining the parameters

4. Let k = n, $f_i = 1$ for $i = 1$ to k.
5. If server k is marked as a bottleneck, $f_i \leftarrow y_k * f_i$ for $i = 1$ to $k - 1$.

   Otherwise, $f_k \leftarrow x_k * f_k$.
6. Stop if $k = 2$. Otherwise, let $k = k - 1$, go to 5.

Theorem 6.1: Procedure 6.2 gives the total queueing time of all servers, where the queueing time of each server is obtained from Procedure 6.1.

Proof:

(i) When n = 1 (i.e. the system has only one single server), based on the 4th and 5th step in Procedure 6.2, f_1 is 1, which correctly describes this single server system queueing time.

(ii) When n = 2 (i.e. adding the 2nd server to the single server system), the 2nd server can be either a bottleneck or a non-bottleneck.

If it is a bottleneck, its queueing time is $QT_2 = \alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} - (1-y_2)QT_1$. Total system queueing time is $\sum_{i=1}^{2} QT_i = QT_1 + QT_2 = \alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + y_2 QT_1 = \alpha_n\left(\frac{\rho_n}{1-\rho_n}\right)\frac{1}{\mu_n} + y_n\sum_{i=1}^{n-1} QT_i$.

If it is a non-bottleneck, its queueing time is $QT_2 = x_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2}$. Total system queueing time is $\sum_{i=1}^{2} QT_i = QT_1 + QT_2 = QT_1 + x_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} = \sum_{i=1}^{n-1} QT_i + x_n\alpha_n\left(\frac{\rho_n}{1-\rho_n}\right)\frac{1}{\mu_n}$.

(iii) When n = k, the k-th server can be either a bottleneck or a non-bottleneck.

If it is a bottleneck, based on Procedure 6.1, its queueing time is $QT_k = \alpha_k\left(\frac{\rho_k}{1-\rho_k}\right)\frac{1}{\mu_k} - (1-y_k)\sum_{i=1}^{k-1} QT_i,$. Total system queueing time is

$\sum_{i=1}^{k} QT_i = \sum_{i=1}^{k-1} QT_i + QT_k = \alpha_k \left(\frac{\rho_k}{1-\rho_k} \right) \frac{1}{\mu_k} + y_k \sum_{i=1}^{k-1} QT_i$. A weight y_k is given to all the servers in front of the k-th server as described in Procedure 6.2.

If it is a non-bottleneck, its queueing time is $QT_k = x_k \alpha_k \left(\frac{\rho_k}{1-\rho_k} \right) \frac{1}{\mu_k}$. Total system queueing is $\sum_{i=1}^{k} QT_i = \sum_{i=1}^{k-1} QT_i + QT_k = \sum_{i=1}^{k-1} QT_i + x_k \alpha_k \left(\frac{\rho_k}{1-\rho_k} \right) \frac{1}{\mu_k}$. A weight x_k is given to the k-th server as described in Procedure 6.2.

(iv) When n = k + 1, the (k + 1)-th server can be either a bottleneck or a non-bottleneck.

If it is a bottleneck, based on Procedure 6.1, its queueing time is $QT_{k+1} = \alpha_{k+1} \left(\frac{\rho_{k+1}}{1-\rho_{k+1}} \right) \frac{1}{\mu_{k+1}} - (1-y_{k+1}) \sum_{i=1}^{k} QT_i$,. Total system queueing time is $\sum_{i=1}^{k+1} QT_i = \sum_{i=1}^{k} QT_i + QT_{k+1} = \alpha_{k+1} \left(\frac{\rho_{k+1}}{1-\rho_{k+1}} \right) \frac{1}{\mu_{k+1}} + y_{k+1} \sum_{i=1}^{k} QT_i$. A weight y_{k+1} is given to all the servers in front of the k-th server as described in Procedure 6.2.

If it is a non-bottleneck, its queueing time is $QT_{k+1} = x_{k+1} \alpha_{k+1} \left(\frac{\rho_{k+1}}{1-\rho_{k+1}} \right) \frac{1}{\mu_{k+1}}$. Total system queueing is $\sum_{i=1}^{k+1} QT_i = \sum_{i=1}^{k} QT_i + QT_{k+1} = \sum_{i=1}^{k} QT_i + x_{k+1} \alpha_{k+1} \left(\frac{\rho_{k+1}}{1-\rho_{k+1}} \right) \frac{1}{\mu_{k+1}}$. A weight x_{k+1} is given to the (k+1)-th server as described in Procedure 6.2. Q.E.D.

It should be noted that α_i is the variability of the i-th server in its ASIA system, not in its original system (except for the first server, where they are the same). Eq. (6.26) describes the contribution (determining by f_i) of each server's ASIA system queueing time to the system queueing time. Furthermore, the value of f_i always equals 1 for the system bottleneck, which means unit weight is always given to the system bottleneck. However, there will be a weight on all other servers' ASIA system queueing times. As we have seen in Chapter 5, if all service time SCVs are smaller than 1 with Poisson

arrivals, the contribution factor will be smaller than 1 (and behaves like a discount factor), since both x_k and y_k are smaller than 1. In this situation, reducing the bottleneck service time SCV brings greater improvement on system cycle time than reducing the non-bottleneck service time SCV. On the other hand, when the arrival process is Poisson, if all service time SCVs are greater than 1, the contribution factor can be greater than 1.

6.4 Implementation of Approximate Models

Procedure 6.1 gives us a way to approximate the queueing time of each server in single-server tandem queues. The question becomes how to approximate the parameters (x_i or y_i) accurately. We will begin with a conventional approach, which assumes service time SCV values are known. In Section 6.4.2, we relax this assumption, and propose a more practical approach.

6.4.1 Approximate Model based on Parametric-Decomposition Approaches

In Section 5.8.2, we presented two heuristics to approximate the intrinsic ratios of STQB. In Section 5.8.3, we extended the second heuristic to approximate the intrinsic ratios of STQF. In both sections, the service time SCVs are smaller than 1. In Section 5.8.4, we extended the models in Section 5.8.2 to approximate the intrinsic ratios of STQB when the service time SCVs are greater than 1. In the present section, inspired by Observation 5.1, 5.3 and 5.4, we use the second heuristic from Section 5.8.2 to approximate the parameters in Procedure 6.1. From Observation 5.1, 5.3 and 5.4, we know that the parametric-decomposition approach performs well at some specific points when the arrival process is Poisson. Specifically, the parametric-decomposition approach performs well around 70 ~ 80% in STQB and around 80 ~ 99.9% in STQF. Since the utilizations of those points are pretty high, we can have a good estimate of intrinsic ratio

at high utilization. Due to the nearly-linear relationship, the single point intrinsic ratio based approximate model can give us a good estimate of queueing time in heavy traffic.

It would be desirable to approximate the intrinsic ratios based on two point linear interpolation or extrapolation, since the slope of intrinsic ratios may not be close to 0. However, the parametric-decomposition approach can give us only one good estimate of intrinsic ratio. Using the best estimate of the intrinsic ratio at one accurate point may be better than finding two points, in which one of them is not very accurate.

Furthermore, as we have observed in Section 5.8.2 and Figure 6.1 (more examples will be given in Section 6.5), the slope of intrinsic ratios is close to zero when service time SCV is smaller than 1 and the initial arrival process is Poisson. In this case, the single point intrinsic ratio based approximate models seem to suffice.

Therefore, when the intrinsic ratio is determined by QNA, the approximate models (i.e. Procedure 6.3) are designed based on the single point approach. The performance of Procedure 6.3 is tested by several examples in Section 6.5.

Procedure 6.3 (Intrinsic Ratio Method based on QNA):

1. Determine the queueing model of the i-th server based on Procedure 6.1, where i increases from 1 to n. Let $k = 2$.
2. If the unknown variable in the queueing model is x_k, go to a. If it is y_k, go to b.
    a. If $ST_{BN_k} = ST_k$, compute x_k using QNA at 80% system utilization, where BN_k is the immediate bottleneck in front of server k. Otherwise, compute x_k using QNA at 99.9% system utilization.
    b. If $ST_{k-1} > (2/3)\ ST_k$, compute y_k using QNA at 80% system utilization. Otherwise, compute y_k using QNA at 70% system utilization.

    If $k = n$, go to 3. Otherwise, let $k = k + 1$, then go to 2.
3. Use x_i (or y_i) to approximate the queueing times of the i-th server at other traffic intensity.

The chosen of parameters in 2.a is motivated by Observation 5.3, and the chosen of parameters in 2.b (e.g. 2/3) is motivated by Observation 5.1 and 5.4. More discussion will be given in Section 5.9.

6.4.2 Approximate Model based on Historical Queueing Times

In Procedure 6.3, we use the parametric-decomposition method to calculate the queueing times at specific traffic intensity. In practice, we usually determine the service time SCV based on historical data. However, if we have to analyze the historical data, we may also obtain the mean queueing time at the same time. A new approximate model is developed based on historical queueing times instead of the parametric-decomposition approach. Depending on the accessibility of historical data, we may get queueing time performance at one or two different utilization levels. The procedures are as follows:

Procedure 6.4 (Single-Point Intrinsic Ratio Method with Historical Data):

1. Find the historical queueing time of each server at a specific utilization.
2. Based on the equations from Procedure 6.1, calculate x_i (or y_i) of the i-th server, where i increases from 1 to n.
3. Use x_i (or y_i) to approximate the queueing times at other traffic intensity.

Procedure 6.4a (Two-Point Intrinsic Ratio Method with Historical Data):

1. Find the historical queueing times of each server at two specific utilizations.
2. Based on the equations from Procedure 6.1, calculate x_i (or y_i) of the i-th server, where i increases from 1 to n, at the two utilizations.
3. Extrapolate (or interpolate) x_i (or y_i) at other utilizations, and use it to approximate the queueing times at other traffic intensity.

By using Procedure 6.4 and 6.4a, we can approximate the intrinsic ratio directly, instead of Procedure 6.3, which relies on the Observation 5.1, 5.3 and 5.4. However, due to Procedure 6.1, Procedure 6.4 and 6.4a still need the variability of each server in its ASIA system.

6.5 Performance of Approximate Models

The testing of the approximate models introduced in Sections 6.3 and 6.4 is conducted in two parts. In the first part, we compared the intrinsic ratio (IR) approach with the previous approximate models in the literature.

In the second part, we conduct simulations on five single servers in series with different combinations of service time SCV and bottleneck locations. Both Procedure 6.3 and 6.4 will be compared with QNA and QNET. The underlying structure (i.e. nearly-linear relationship of the intrinsic ratio) of queues in series is deserved in these tests.

6.5.1 Comparison with Previous Work

Procedure 6.3 is developed based on the observations from Poisson arrivals at the first server. In this section, we test the performance of Procedure 6.3 in more general setting, where the initial arrival process is not Poisson.

The following six cases, from Case A-1 to A-6, were first presented in Suresh and Whitt (1990b) and then Dai, Nguyen and Reiman (1994). There are nine serial servers in Case A-1 and A-2, and ten serial servers in Case A-3 to A-6. The service time distribution is deterministic if SCV is 0, exponential if SCV is 1, and hyperexponential if SCV is 8. In their experiments, each expected queueing time is obtained from 10 replications of 30,000 arrivals after discarding the first 2,000 data points.

Based on their results, we can compare the performances of Procedure 6.3, QNA, QNET and Sequential Bottleneck Decomposition (SBD). Except for the last two columns, all the data in Table 6.3 ~ 6.8 are directly cited from Dai, Nguyen and Reiman (1994).

To compare the performance of each approximate model objectively, we did not compare the total queueing time percentage errors, since an unreliable method could give small net percentage errors if its negative errors for some servers are compensated by positive errors for others. Instead, we compare the total absolute error percentage which is the ratio of the total absolute error to the total system queueing times from simulation, where the total absolute error is the summation of the absolute differences between the simulation queueing times and the approximated queuing times. We also compare the system bottleneck errors among the four methods. The smallest ones are highlighted by bold lines in Table 6.3 ~ 6.8.

Case A-1 (Nine single queues in series):

- Inter-arrival time distribution: Deterministic with mean = 1 and $C_{a1}^2 = 0$,
- Service time mean: (0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.9)
- Service time SCV: (1, 1, 1, 1, 1, 1, 1, 1, 1)

Table 6.3 Queueing time approximations of nine servers in series in Case A-1

Station Number	Simulation		QNA		QNET		SBD		IR Method	
	QT	90% CI	QT	Error %	QT	Error %	QT	Error %	QT	Error %
1	0.290	2.41%	0.45	55.17%	0.45	55.17%	0.45	55.17%	0.45	55.17%
2	0.491	1.43%	0.61	24.64%	0.66	34.42%	0.66	34.42%	0.58	17.72%
3	0.607	1.32%	0.72	17.90%	0.74	21.91%	0.74	21.91%	0.67	10.31%
4	0.666	1.20%	0.78	17.42%	0.79	18.62%	0.79	18.62%	0.74	10.38%
5	0.706	1.42%	0.82	16.79%	0.82	16.15%	0.82	16.15%	0.78	10.77%
6	0.731	1.78%	0.85	16.51%	0.84	14.91%	0.84	14.91%	0.82	11.57%
7	0.748	1.34%	0.87	16.19%	0.85	13.64%	0.85	13.64%	0.84	12.25%
8	0.775	1.68%	0.88	13.58%	0.86	10.97%	0.86	10.97%	0.86	10.55%
9	5.031	4.31%	7.99	58.74%	6.97	38.54%	4.05	-19.50%	5.68	**12.99%**
Total	10.045		13.97	39.09%	12.98	29.22%	10.06	0.15%	11.41	13.60%
Total absolute error			3.93		2.94		1.98		1.37	
Total absolute % error			39.09%		29.22%		19.68%		**13.60%**	

In Case A-1, the first bottleneck is station 9 and the second bottleneck is station 1. Based on Procedure 6.3, since $ST_1 = ST_i$ for $i = 2$ to 8, x_i is obtained by QNA at 80% system utilization. Because $ST_8 = (2/3)\ ST_9$, y_9 is obtained by QNA at 70% system utilization.

The half-width of the 90% confidence interval is shown in the third column next to the simulated queueing times. For each method, the percentage error (compared with simulations) is in the column to the right of the approximate value. The IR method gives the best approximation among the four.

Case A-2 (Nine single queues in series):

- Inter-arrival time distribution: Hyperexponential with mean = 1 and $C_{a1}^2 = 8$,
- Service time mean: (0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.9)
- Service time SCV: (1, 1, 1, 1, 1, 1, 1, 1, 1)

Table 6.4 Queueing time approximations of nine servers in series in Case A-2

Station Number	Simulation		QNA		QNET		SBD		IR Method	
	QT	90% CI	QT	Error %	QT	Error %	QT	Error %	QT	Error %
1	3.284	3.50%	4.05	23.33%	4.05	23.33%	4.05	23.33%	4.05	23.33%
2	2.321	4.18%	2.92	25.64%	1.81	-22.02%	1.82	-21.59%	3.15	35.89%
3	1.914	3.40%	2.19	14.43%	1.47	-23.20%	1.49	-22.15%	2.51	31.29%
4	1.719	4.07%	1.73	0.39%	1.16	-32.52%	1.19	-30.77%	2.05	19.49%
5	1.598	3.69%	1.43	-10.61%	1.07	-33.04%	1.10	-31.16%	1.73	8.00%
6	1.478	4.13%	1.24	-16.22%	1.03	-30.31%	1.06	-28.28%	1.49	0.87%
7	1.423	3.23%	1.12	-21.54%	1.00	-29.73%	1.03	-27.62%	1.32	-7.04%
8	1.413	4.67%	1.04	-26.50%	0.98	-30.64%	1.01	-28.52%	1.20	-14.89%
9	30.116	16.84%	8.90	-70.45%	6.04	-79.94%	36.45	21.03%	26.79	**-11.05%**
Total	45.266		24.60	-45.65%	18.61	-58.89%	49.20	8.69%	44.30	-2.13%
Total absolute error			23.95		28.19		10.27		6.31	
Total absolute % error			52.91%		62.27%		22.68%		**13.95%**	

Case A-2 is similar to the settings of Case A-1, but with higher arrival process variability. The IR method outperforms the other methods.

Case A-3 (Ten single queues in series):

- Inter-arrival time distribution: Hyperexponential with mean = 1 and $C_{a1}^2 = 8$,

- Service time mean: (0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.9)

- Service time SCV: (0, 1, 1, 1, 1, 1, 1, 1, 1, 1)

Table 6.5 Queueing time approximations of ten servers in series in Case A-3

Station Number	Simulation		QNA		QNET		SBD		IR Method	
	QT	90% CI	QT	Error %	QT	Error %	QT	Error %	QT	Error %
1	2.441	3.69%	3.60	47.48%	3.60	47.48%	3.60	47.48%	3.60	47.48%
2	1.796	3.90%	2.75	53.34%	0.79	-56.01%	0.80	-55.46%	3.03	68.49%
3	2.008	4.38%	2.09	3.91%	1.32	-34.26%	1.34	-33.27%	2.42	20.58%
4	1.804	3.32%	1.66	-8.02%	1.25	-30.71%	1.27	-29.60%	1.99	10.23%
5	1.663	4.15%	1.39	-16.66%	1.13	-32.05%	1.15	-30.85%	1.68	0.96%
6	1.562	3.65%	1.21	-22.47%	1.06	-32.14%	1.08	-30.86%	1.46	-6.70%
7	1.449	3.80%	1.10	-24.15%	1.01	-30.30%	1.04	-28.23%	1.30	-10.36%
8	1.405	3.27%	1.03	-26.88%	0.98	-30.25%	1.01	-28.11%	1.19	-15.63%
8	1.398	4.72%	0.98	-29.79%	0.96	-31.33%	0.99	-29.18%	1.10	-21.02%
10	29.970	16.90%	8.57	-71.41%	5.14	-82.85%	36.45	21.62%	26.80	**-10.58%**
Total	45.496		24.37	-46.42%	17.24	-62.11%	48.73	7.11%	44.56	-2.06%
Total absolute error			25.51		30.57		12.04		6.94	
Total absolute % error			56.08%		67.20%		26.47%		**15.26%**	

In Case A-3, the first bottleneck is station 10, and the second bottleneck is station 1. Based on Procedure 6.3, since $ST_1 = ST_i$ for i = 2 to 9, x_i is obtained by QNA at 80% utilization. Because $ST_9 = (2/3)\ ST_{10}$, y_{10} is obtained by QNA at 70% utilization. The IR method outperforms the others.

Case A-4 (Ten single queues in series):

- Inter-arrival time distribution: Hyperexponential with mean = 1 and $C_{a1}^2 = 8$,

- Service time mean: (0.9, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.9)

- Service time SCV: (0, 1, 1, 1, 1, 1, 1, 1, 1, 1)

Table 6.6 Queueing time approximations of ten servers in series in Case A-4

Station Number	Simulation		QNA		QNET		SBD		IR Method	
	QT	90% CI	QT	Error %	QT	Error %	QT	Error %	QT	Error %
1	32.780	15.61%	32.40	-1.16%	32.40	-1.16%	32.40	-1.16%	32.40	-1.16%
2	0.418	2.63%	1.13	171.29%	0.49	17.22%	0.45	7.66%	0.46	9.38%
3	0.674	1.93%	1.05	55.75%	0.82	21.66%	0.66	-2.08%	0.65	-3.03%
4	0.800	1.75%	1.00	24.48%	0.87	8.75%	0.74	-7.50%	0.76	-4.64%
5	0.860	1.98%	0.96	11.78%	0.88	2.33%	0.79	-8.14%	0.82	-4.22%
6	0.908	1.76%	0.94	3.44%	0.89	-1.98%	0.82	-9.69%	0.86	-5.56%
7	0.906	1.88%	0.93	2.11%	0.89	-1.77%	0.84	-7.28%	0.88	-3.27%
8	0.921	1.95%	0.92	-0.53%	0.89	-3.37%	0.85	-7.71%	0.89	-3.71%
8	0.940	2.45%	0.91	-3.16%	0.90	-4.26%	0.86	-8.51%	0.89	-5.03%
10	14.039	13.56%	8.16	-41.88%	8.28	-41.02%	5.46	-61.11%	9.85	**-29.85%**
Total	53.246		48.39	-9.12%	47.31	-11.15%	43.87	-17.61%	48.46	-8.99%
Total absolute error			7.73		6.55		9.44		4.87	
Total absolute % error			14.52%		12.31%		17.73%		**9.14%**	

In Case A-4, the bottleneck is station 1. Based on Procedure 6.3, since $ST_1 > ST_i$ for i = 2 to 9, x_i is obtained by QNA at 99.9% utilization. Because $ST_1 = ST_{10}$, x_{10} is obtained by QNA at 80% utilization. The IR method outperforms the others.

Case A-5 (Ten single queues in series):

- Inter-arrival time distribution: Deterministic with mean = 1 and $C_{a1}^2 = 0$,

- Service time mean: (0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.9)

- Service time SCV: (8, 1, 1, 1, 1, 1, 1, 1, 1, 1)

Table 6.7 Queueing time approximations of ten servers in series in Case A-5

Station Number	Simulation		QNA		QNET		SBD		IR Method	
	QT	90% CI	QT	Error %	QT	Error %	QT	Error %	QT	Error %
1	3.521	3.83%	3.60	2.24%	3.60	2.24%	3.60	2.24%	3.60	2.24%
2	1.873	3.36%	1.75	-6.78%	2.44	30.27%	2.44	30.27%	1.47	-21.30%
3	1.351	2.15%	1.44	6.69%	1.16	-14.14%	1.16	-14.14%	1.31	-2.98%
4	1.227	3.10%	1.25	1.59%	1.03	-16.06%	1.03	-16.06%	1.19	-2.70%
5	1.185	2.19%	1.12	-5.34%	0.98	-17.30%	0.98	-17.30%	1.11	-6.30%
6	1.148	1.83%	1.04	-9.24%	0.95	-17.25%	0.95	-17.25%	1.05	-8.49%
7	1.094	3.11%	0.99	-9.43%	0.94	-14.08%	0.93	-14.99%	1.01	-7.89%
8	1.068	3.00%	0.96	-10.29%	0.92	-13.86%	0.92	-13.86%	0.98	-8.52%
8	1.041	2.02%	0.94	-9.97%	0.92	-11.62%	0.92	-11.62%	0.96	-8.25%
10	8.596	3.66%	8.31	**-3.28%**	8.07	-6.12%	4.05	-52.89%	6.17	-28.25%
Total	22.104		21.40	-3.19%	21.01	-4.95%	16.98	-23.18%	18.85	-14.73%
Total absolute error			1.08		2.39		6.42		3.41	
Total absolute % error			**4.90%**		10.79%		29.03%		15.45%	

The queueing time approximate error is more critical when queueing time is longer. If queueing time is short, large queueing time approximation errors only induce small approximation errors of total system cycle time, since they will be mitigated by the service times.

In Case A-5, the first bottleneck is station 10, and the second bottleneck is station 1. Based on Procedure 6.3, since $ST_1 = ST_i$ for i = 2 to 9, x_i is obtained by QNA at 80% utilization. Because $ST_9 = (2/3)\ ST_{10}$, y_{10} is obtained by QNA at 70% utilization. Although QNA outperforms the others, the errors are less important in this situation, since queueing times are relatively short compared to the service times.

Case A-6 (Ten single queues in series):

- Inter-arrival time distribution: Deterministic with mean = 1 and $C_{a1}^2 = 0$,
- Service time mean: (0.9, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.9)
- Service time SCV: (8, 1, 1, 1, 1, 1, 1, 1, 1, 1)

Table 6.8 Queueing time approximations of ten servers in series in Case A-6

Station Number	Simulation		QNA		QNET		SBD		IR Method	
	QT	90% CI	QT	Error %	QT	Error %	QT	Error %	QT	Error %
1	29.548	5.27%	32.40	9.65%	32.40	9.65%	32.40	9.65%	32.40	9.65%
2	3.210	3.21%	3.37	4.86%	3.25	1.25%	4.05	26.17%	4.04	25.94%
3	2.022	3.56%	2.48	22.56%	1.42	-29.77%	1.82	-9.99%	2.65	31.00%
4	1.787	3.36%	1.91	6.89%	1.12	-37.33%	1.49	-16.62%	1.87	4.82%
5	1.582	4.24%	1.55	-2.25%	1.04	-34.26%	1.19	-24.78%	1.44	-8.88%
6	1.496	2.27%	1.31	-12.18%	1.00	-33.16%	1.10	-26.47%	1.20	-19.70%
7	1.443	3.26%	1.16	-19.28%	0.98	-32.09%	1.06	-26.54%	1.07	-26.01%
8	1.351	2.58%	1.07	-20.84%	0.96	-28.94%	1.03	-23.76%	0.99	-26.48%
8	1.318	2.50%	1.01	-23.49%	0.95	-27.92%	1.01	-23.37%	0.95	-27.78%
10	16.360	5.71%	8.72	-46.67%	8.12	-50.37%	24.18	47.80%	9.76	**-40.36%**
Total	60.117		54.98	-8.54%	51.24	-14.77%	69.33	15.33%	56.38	-6.22%
Total absolute error			12.31		14.66		13.81		12.53	
Total absolute % error			**20.48%**		24.39%		22.97%		**20.85%**	

In Case A-6, the single bottleneck is station 1. Based on Procedure 6.3, since $ST_1 > ST_i$ for i = 2 to 9, x_i is obtained by QNA at 99.9% utilization. Because $ST_1 = ST_{10}$, x_{10} is obtained by QNA at 80% utilization. The IR method and QNA outperform the others.

In these six cases, the IR method performs the best in 4 cases. The IR method and QNA perform equally well and outperform QNET and SBD in 1 case. The QNA performs the best in 1 case.

6.5.2 Performance for Five Single-Server Queues in Series

In this section, Procedure 6.3 and 6.4 will be tested by ten cases composed of 5 servers in series with no job recirculation inside the system. The first server is the only server which is fed by the exogenous arrival process as shown in Figure 6.4.

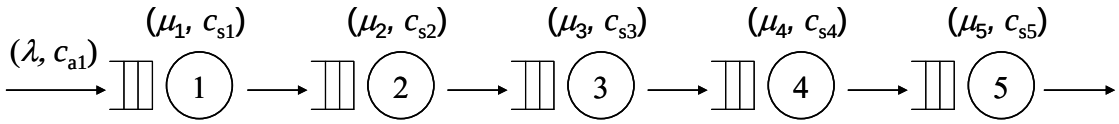

Figure 6.4 Five single-server queues in series

In each case, the service time SCV varies. The bottleneck can be the first, third or fifth server. The mean bottleneck service time is always 30. The service times are independently and identically distributed and follow a gamma distribution. The initial arrival process is Poisson in most cases. Traffic intensity of the bottleneck varies from 10% ~ 95%. In total, 100 replications are conducted for each observation. Each replication is composed of 200,000 to 400,000 data points after a 50 year warm-up period (i.e. 87,600 (at 10%) to 832,200 (at 95%) data points are discarded).

The performance of Procedure 6.3 (IR Method with QNA) and Procedure 6.4 (IR Method with Historical Data) are compared with QNA and QNET. The first case has been shown in Section 6.2, where it is used to demonstrate the underlying structure of intrinsic ratios in tandem queues.

Case B-1:

- Poisson arrivals

- Service time mean: (20, 25, 30, 25, 20) and SCV: (0.5, 0.5, 0.5, 0.5, 0.5)

Table 6.9 Queueing time approximations in Case B-1

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	1.07	1.49	2.11	1.27	0.73	6.67	0.22%	0.21%	0.17%	0.19%	0.23%		
	20%	2.31	3.27	4.74	2.76	1.56	14.63	0.18%	0.19%	0.15%	0.17%	0.18%		
	30%	3.75	5.45	8.11	4.55	2.48	24.33	0.14%	0.16%	0.15%	0.16%	0.17%		
	40%	5.46	8.16	12.59	6.75	3.57	36.53	0.13%	0.16%	0.17%	0.16%	0.17%		
	50%	7.51	11.66	18.92	9.54	4.84	52.47	0.14%	0.17%	0.16%	0.18%	0.17%		
	60%	10.01	16.37	28.57	13.23	6.37	74.55	0.17%	0.18%	0.22%	0.17%	0.16%		
	70%	13.14	22.96	44.90	18.30	8.25	107.55	0.15%	0.17%	0.27%	0.21%	0.16%		
	80%	17.17	33.06	78.57	25.72	10.62	165.14	0.20%	0.25%	0.36%	0.22%	0.18%		
	90%	22.50	49.89	183.81	37.60	13.70	307.50	0.20%	0.34%	0.85%	0.24%	0.18%		
	95%	25.86	63.91	398.97	46.59	15.61	550.94	0.20%	0.30%	1.43%	0.31%	0.21%		
QNA	10%	1.07	1.70	2.49	1.69	1.06	8.02	-0.02%	14.29%	18.03%	33.19%	45.16%	20.16%	
	20%	2.31	3.73	5.54	3.65	2.22	17.45	0.05%	13.92%	16.95%	31.94%	42.98%	19.21%	
	30%	3.75	6.17	9.32	5.87	3.46	28.57	0.01%	13.21%	14.99%	29.17%	39.28%	17.41%	
	40%	5.45	9.15	14.13	8.42	4.76	41.91	-0.08%	12.11%	12.24%	24.70%	33.31%	14.76%	
	50%	7.50	12.90	20.51	11.39	6.14	58.43	-0.09%	10.59%	8.38%	19.36%	26.79%	11.38%	
	60%	10.00	17.75	29.59	15.02	7.67	80.03	-0.06%	8.45%	3.56%	13.49%	20.49%	7.37%	
	70%	13.13	24.34	44.03	19.80	9.51	110.81	-0.10%	6.05%	-1.94%	8.20%	15.25%	4.68%	
	80%	17.14	33.94	71.93	26.79	11.88	161.69	-0.14%	2.68%	-8.46%	4.14%	11.86%	5.98%	
	90%	22.50	49.50	153.90	38.50	15.17	279.57	0.01%	-0.78%	-16.27%	2.38%	10.78%	10.63%	
	95%	25.91	61.72	316.85	48.02	17.34	469.85	0.18%	-3.42%	-20.58%	3.07%	11.07%	15.88%	11.96%
QNET	10%	1.07	1.41	1.97	1.25	0.75	6.46	-0.02%	-5.34%	-6.80%	-1.28%	3.24%	3.95%	
	20%	2.31	3.11	4.44	2.76	1.62	14.23	0.05%	-5.08%	-6.34%	-0.19%	4.34%	3.69%	
	30%	3.75	5.18	7.63	4.59	2.63	23.79	0.01%	-4.81%	-5.84%	1.00%	5.93%	3.82%	
	40%	5.45	7.79	11.92	6.88	3.82	35.86	-0.08%	-4.57%	-5.30%	1.90%	6.96%	3.89%	
	50%	7.50	11.15	17.97	9.80	5.23	51.65	-0.09%	-4.40%	-5.05%	2.74%	8.03%	4.05%	
	60%	10.00	15.64	27.11	13.67	6.94	73.36	-0.06%	-4.46%	-5.11%	3.28%	8.98%	4.29%	
	70%	13.13	21.93	42.47	19.00	9.06	105.58	-0.10%	-4.48%	-5.41%	3.83%	9.76%	4.63%	
	80%	17.14	31.37	73.40	26.80	11.73	160.44	-0.14%	-5.11%	-6.58%	4.18%	10.44%	5.49%	
	90%	22.50	47.13	166.82	39.28	15.23	290.95	0.01%	-5.53%	-9.24%	4.45%	11.18%	7.47%	
	95%	25.91	59.71	358.41	49.02	17.42	510.47	0.18%	-6.57%	-10.17%	5.22%	11.57%	8.90%	7.05%
IR Method with QNA	10%	1.07	1.48	1.60	1.14	0.71	6.00	-0.02%	-0.47%	-24.34%	-10.56%	-2.31%	10.07%	
	20%	2.31	3.27	3.65	2.50	1.54	13.27	0.05%	-0.02%	-22.93%	-9.51%	-1.09%	9.35%	
	30%	3.75	5.47	6.38	4.17	2.50	22.27	0.01%	0.46%	-21.29%	-8.34%	0.65%	8.82%	
	40%	5.45	8.24	10.15	6.25	3.64	33.74	-0.08%	0.98%	-19.33%	-7.39%	1.91%	8.44%	
	50%	7.50	11.84	15.66	8.93	5.00	48.93	-0.09%	1.50%	-17.25%	-6.41%	3.29%	8.04%	
	60%	10.00	16.68	24.31	12.50	6.67	70.16	-0.06%	1.89%	-14.90%	-5.54%	4.67%	7.51%	
	70%	13.13	23.53	39.53	17.50	8.75	102.44	-0.10%	2.49%	-11.96%	-4.37%	6.05%	6.74%	
	80%	17.14	33.94	71.93	25.00	11.43	159.45	-0.14%	2.68%	-8.46%	-2.79%	7.58%	5.50%	
	90%	22.50	51.58	176.29	37.51	15.00	302.88	0.01%	3.40%	-4.09%	-0.25%	9.51%	3.45%	
	95%	25.91	65.88	395.03	47.51	17.27	551.59	0.18%	3.08%	-0.99%	1.97%	10.63%	1.55%	3.92%
Single-Point (80%) IR Method w/ Historical Data	10%	1.07	1.43	1.93	1.17	0.66	6.26	-0.02%	-4.11%	-8.48%	-7.98%	-9.19%	6.13%	
	20%	2.31	3.15	4.38	2.57	1.43	13.85	0.05%	-3.59%	-7.50%	-6.91%	-8.06%	5.40%	
	30%	3.75	5.28	7.59	4.29	2.32	23.23	0.01%	-3.02%	-6.40%	-5.71%	-6.44%	4.53%	
	40%	5.45	7.97	11.95	6.43	3.38	35.18	-0.08%	-2.41%	-5.10%	-4.73%	-5.27%	3.69%	
	50%	7.50	11.46	18.19	9.19	4.65	50.98	-0.09%	-1.75%	-3.90%	-3.71%	-3.99%	2.85%	
	60%	10.00	16.17	27.79	12.86	6.20	73.02	-0.06%	-1.21%	-2.71%	-2.82%	-2.71%	2.04%	
	70%	13.13	22.86	44.31	18.01	8.13	106.44	-0.10%	-0.40%	-1.32%	-1.61%	-1.42%	1.03%	
	80%	17.17	33.07	78.57	25.72	10.62	165.15	0.00%	0.04%	-0.01%	0.00%	0.01%	0.01%	
	90%	22.50	50.44	185.90	38.59	13.94	311.38	0.01%	1.11%	1.14%	2.62%	1.79%	1.26%	
	95%	25.91	64.56	406.91	48.88	16.06	562.32	0.18%	1.03%	1.99%	4.91%	2.84%	2.06%	1.72%
Two-Point (70/80%) IR Method w/ Historical Data	10%	1.07	1.48	2.16	1.29	0.73	6.74	-0.02%	-0.35%	2.29%	1.76%	0.49%	1.20%	
	20%	2.31	3.26	4.81	2.81	1.56	14.74	0.05%	-0.43%	1.45%	1.54%	0.35%	0.90%	
	30%	3.75	5.42	8.17	4.61	2.50	24.46	0.01%	-0.45%	0.80%	1.43%	0.69%	0.70%	
	40%	5.45	8.13	12.64	6.82	3.59	36.63	-0.08%	-0.41%	0.40%	1.04%	0.50%	0.48%	
	50%	7.50	11.63	18.92	9.60	4.86	52.51	-0.09%	-0.31%	-0.03%	0.66%	0.40%	0.25%	
	60%	10.00	16.32	28.47	13.25	6.39	74.42	-0.06%	-0.29%	-0.36%	0.12%	0.26%	0.25%	
	70%	13.14	22.96	44.77	18.28	8.26	107.40	0.00%	0.01%	-0.30%	-0.12%	0.08%	0.15%	
	80%	17.17	33.07	78.57	25.72	10.62	165.15	0.00%	0.04%	-0.01%	0.00%	0.01%	0.01%	
	90%	22.50	50.28	184.96	38.00	13.73	309.47	0.01%	0.78%	0.63%	1.07%	0.24%	0.64%	
	95%	25.91	64.27	405.16	47.77	15.69	558.80	0.18%	0.57%	1.55%	2.53%	0.48%	1.43%	0.81%

The queueing time in the first section of the table is from the simulation results. The percentage on the right is the half-width of 90% confidence intervals. The other sections of the table contain results from different approximate models, which are specified in the very left column. The percentage error for each approximation (compared with simulation) is shown in the corresponding column on the right. The average (absolute) percentage error is calculated based on the total *absolute* difference (between approximations and simulation results) divided by the system queueing time from simulations at each specific utilization. The total average percentage error (on the very right column) is computed in the same manner (based on the *absolute* differences), considering all utilizations. When the arrival process is Poisson, P-K formula gives exact results, and thus QNA, QNET and IR methods all give the same exact solutions.

In the IR method with historical data, we use the queueing time at 70% and 80% utilizations as the historical data. Specifically, we assume the historical data are available at 80% utilization for the single-point approach and assume the historical data are available at both 70% and 80% utilizations for the two-point approach. In addition to the simulation results at ten utilization levels, which represent the performances in the future, we rerun the simulations at 70% and 80% utilizations with different random seeds to generate comparable "historical" data.

In Case B-1, the two IR methods with historical data (Procedure 6.4) perform the best, and the IR method with QNA (Procedure 6.3) is the second. It should be noted all intrinsic ratio approaches perform well in heavy traffic, but QNA and QNET do not. QNA performs well at 70% utilization and QNET performs well in light traffic.

Case B-2:

- Poisson arrivals

- Service time mean: (25, 28, 30, 20, 25) and SCV: (0.25, 0.25, 0.25, 0.25, 0.25)

Table 6.10 Queueing time approximations in Case B-2

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	1.42	1.23	1.29	0.33	0.73	5.00	0.16%	0.16%	0.18%	0.23%	0.18%		
	20%	3.13	2.74	2.87	0.70	1.58	11.01	0.11%	0.13%	0.13%	0.18%	0.13%		
	30%	5.20	4.62	4.89	1.11	2.57	18.39	0.14%	0.15%	0.16%	0.20%	0.16%		
	40%	7.80	7.08	7.60	1.56	3.76	27.81	0.13%	0.16%	0.15%	0.18%	0.16%		
	50%	11.15	10.41	11.41	2.09	5.25	40.33	0.15%	0.17%	0.16%	0.16%	0.17%		
	60%	15.65	15.26	17.25	2.71	7.18	58.05	0.17%	0.19%	0.22%	0.17%	0.18%		
	70%	21.89	22.85	27.25	3.45	9.73	85.17	0.17%	0.25%	0.27%	0.14%	0.16%		
	80%	31.27	36.43	48.29	4.35	13.32	133.66	0.20%	0.31%	0.38%	0.16%	0.23%		
	90%	46.89	67.95	118.95	5.51	18.93	258.23	0.26%	0.39%	0.78%	0.15%	0.24%		
	95%	59.27	104.94	276.87	6.23	22.96	470.26	0.27%	0.52%	1.17%	0.15%	0.20%		
QNA	10%	1.42	1.79	2.06	0.88	1.40	7.55	0.05%	45.70%	60.57%	163.50%	90.88%	51.02%	
	20%	3.13	3.95	4.51	1.81	2.91	16.31	-0.07%	44.16%	57.13%	158.64%	84.64%	48.08%	
	30%	5.21	6.55	7.38	2.72	4.44	26.30	0.10%	41.67%	50.79%	146.41%	73.13%	43.02%	
	40%	7.81	9.73	10.74	3.57	5.92	37.77	0.12%	37.44%	41.37%	128.43%	57.35%	35.85%	
	50%	11.16	13.72	14.77	4.32	7.35	51.32	0.08%	31.71%	29.43%	106.48%	39.88%	27.26%	
	60%	15.63	18.93	19.94	4.98	8.84	68.32	-0.16%	24.08%	15.56%	83.81%	23.22%	17.78%	
	70%	21.88	26.25	27.43	5.64	10.73	91.92	-0.05%	14.86%	0.63%	63.72%	10.25%	7.95%	
	80%	31.25	37.82	41.06	6.47	13.69	130.30	-0.06%	3.82%	-14.96%	48.83%	2.73%	8.32%	
	90%	46.88	60.87	80.54	7.78	19.19	215.25	-0.02%	-10.42%	-32.29%	41.09%	1.35%	18.60%	
	95%	59.38	85.43	159.56	8.74	23.92	337.02	0.17%	-18.59%	-42.37%	40.30%	4.18%	29.85%	22.87%
QNET	10%	1.42	1.13	1.19	0.41	0.66	4.81	0.05%	-8.31%	-7.77%	22.85%	-9.57%	6.98%	
	20%	3.13	2.52	2.67	0.88	1.45	10.64	-0.07%	-8.07%	-6.92%	25.33%	-8.08%	6.60%	
	30%	5.21	4.27	4.60	1.41	2.40	17.89	0.10%	-7.67%	-6.11%	27.72%	-6.40%	6.14%	
	40%	7.81	6.54	7.16	2.03	3.58	27.13	0.12%	-7.66%	-5.69%	30.09%	-4.93%	5.90%	
	50%	11.16	9.58	10.75	2.76	5.07	39.33	0.08%	-7.97%	-5.79%	32.15%	-3.56%	5.85%	
	60%	15.63	13.87	16.08	3.63	7.01	56.22	-0.16%	-9.09%	-6.80%	34.10%	-2.32%	6.33%	
	70%	21.88	20.28	24.69	4.69	9.66	81.19	-0.05%	-11.25%	-9.40%	36.13%	-0.77%	7.59%	
	80%	31.25	30.71	40.55	6.00	13.49	122.00	-0.06%	-15.70%	-16.04%	37.92%	1.27%	11.45%	
	90%	46.88	50.08	78.00	7.66	19.63	202.26	-0.02%	-26.29%	-34.43%	39.09%	3.69%	23.89%	
	95%	59.38	67.72	129.04	8.71	24.38	289.22	0.17%	-35.47%	-53.39%	39.89%	6.20%	40.20%	25.49%
IR Method with QNA	10%	1.42	1.18	0.81	0.36	0.57	4.33	0.05%	-4.47%	-37.18%	7.05%	-22.26%	14.39%	
	20%	3.13	2.64	1.85	0.77	1.25	9.64	-0.07%	-3.60%	-35.44%	10.02%	-20.65%	13.75%	
	30%	5.21	4.51	3.26	1.25	2.08	16.31	0.10%	-2.39%	-33.40%	13.08%	-18.80%	12.93%	
	40%	7.81	6.99	5.23	1.82	3.13	24.97	0.12%	-1.31%	-31.16%	16.33%	-16.98%	12.09%	
	50%	11.16	10.40	8.16	2.50	4.46	36.68	0.08%	-0.14%	-28.53%	19.56%	-15.04%	11.11%	
	60%	15.63	15.40	12.88	3.33	6.25	53.49	-0.16%	0.90%	-25.32%	23.03%	-12.92%	10.48%	
	70%	21.88	23.35	21.53	4.38	8.75	79.88	-0.05%	2.19%	-21.00%	27.00%	-10.09%	9.57%	
	80%	31.25	37.82	41.06	5.71	12.50	128.35	-0.06%	3.82%	-14.96%	31.41%	-6.18%	8.10%	
	90%	46.88	71.24	110.72	7.50	18.75	255.09	-0.02%	4.85%	-6.92%	36.11%	-0.97%	5.31%	
	95%	59.38	110.78	272.65	8.64	23.75	475.19	0.17%	5.57%	-1.52%	38.70%	3.47%	2.84%	5.83%
Single-Point (80%) IR Method w/ Historical Data	10%	1.42	1.12	1.08	0.27	0.61	4.50	0.05%	-9.19%	-15.69%	-18.53%	-16.91%	10.02%	
	20%	3.13	2.51	2.47	0.59	1.34	10.03	-0.07%	-8.28%	-14.12%	-16.27%	-15.20%	8.97%	
	30%	5.21	4.30	4.29	0.95	2.23	16.98	0.10%	-7.01%	-12.32%	-13.93%	-13.22%	7.75%	
	40%	7.81	6.67	6.80	1.38	3.34	26.00	0.12%	-5.83%	-10.50%	-11.46%	-11.27%	6.56%	
	50%	11.16	9.94	10.44	1.90	4.77	38.22	0.08%	-4.53%	-8.53%	-9.00%	-9.20%	5.27%	
	60%	15.63	14.76	16.16	2.54	6.68	55.76	-0.16%	-3.30%	-6.33%	-6.36%	-6.94%	3.94%	
	70%	21.88	22.46	26.29	3.33	9.35	83.31	-0.05%	-1.73%	-3.53%	-3.34%	-3.91%	2.19%	
	80%	31.27	36.54	48.30	4.35	13.36	133.81	0.00%	0.29%	0.02%	0.01%	0.27%	0.11%	
	90%	46.88	69.32	122.99	5.71	20.04	264.94	-0.02%	2.02%	3.40%	3.60%	5.84%	2.61%	
	95%	59.38	108.35	290.20	6.57	25.38	489.88	0.17%	3.25%	4.82%	5.56%	10.58%	4.17%	3.39%
Two-Point (70/80%) IR Method w/ Historical Data	10%	1.42	1.26	1.46	0.33	0.77	5.24	0.05%	2.16%	13.41%	0.29%	5.54%	4.82%	
	20%	3.13	2.78	3.17	0.70	1.65	11.42	-0.07%	1.34%	10.48%	0.31%	4.44%	3.74%	
	30%	5.21	4.67	5.28	1.11	2.66	18.92	0.10%	0.91%	7.82%	0.27%	3.54%	2.85%	
	40%	7.81	7.11	7.99	1.57	3.86	28.33	0.12%	0.38%	5.20%	0.22%	2.43%	1.89%	
	50%	11.16	10.41	11.74	2.09	5.32	40.72	0.08%	-0.01%	2.82%	0.00%	1.32%	1.00%	
	60%	15.63	15.19	17.40	2.70	7.20	58.12	-0.16%	-0.42%	0.84%	-0.18%	0.25%	0.44%	
	70%	21.89	22.76	27.19	3.44	9.71	84.99	0.00%	-0.40%	-0.23%	-0.15%	-0.20%	0.21%	
	80%	31.27	36.54	48.30	4.35	13.36	133.81	0.00%	0.29%	0.02%	0.01%	0.27%	0.11%	
	90%	46.88	68.66	120.61	5.52	19.27	260.93	-0.02%	1.05%	1.39%	0.18%	1.75%	1.05%	
	95%	59.38	107.09	284.98	6.25	23.91	481.61	0.17%	2.06%	2.93%	0.34%	4.17%	2.41%	1.51%

Similar to the results of Case B-1, the two IR methods with historical data perform the best, and IR method with QNA performs second best. The average percentage errors of QNA and QNET are both higher than 20%, which are much higher than the errors in Case B-1. This is consistent with our previous observations in Chapter 5: QNA and QNET perform poorly when the service time variability is small.

QNA performs well around 70% ~ 80% utilizations and QNET performs well in light traffic. In heavy traffic, the two-point IR method with historical data performs the best.

Case B-2 possesses the nearly-linear relationship of the intrinsic ratios as shown in Figure 6.5, which explains why the IR method can perform so well.

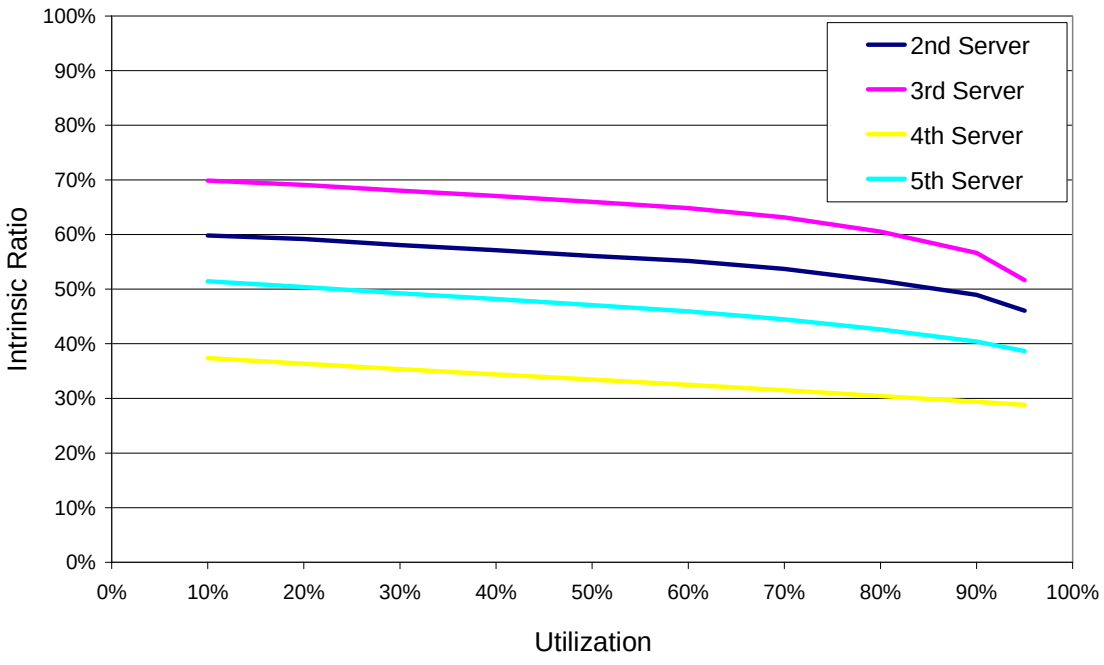

Figure 6.5 Intrinsic Ratios of Case B-2

Case B-3:

- Poisson arrivals

- Service time mean: (20, 23, 25, 28, 30) and SCV: (0.25, 0.25, 0.25, 0.25, 0.25)

Table 6.11 Queueing time approximations in Case B-3

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	0.89	0.83	0.89	1.11	1.23	4.95	0.20%	0.19%	0.22%	0.21%	0.22%		
	20%	1.92	1.81	1.95	2.46	2.74	10.88	0.17%	0.15%	0.17%	0.16%	0.16%		
	30%	3.12	2.98	3.24	4.16	4.68	18.18	0.14%	0.15%	0.15%	0.19%	0.15%		
	40%	4.55	4.40	4.85	6.37	7.26	27.42	0.14%	0.17%	0.12%	0.14%	0.19%		
	50%	6.26	6.19	6.93	9.37	10.92	39.67	0.13%	0.15%	0.17%	0.18%	0.17%		
	60%	8.33	8.51	9.73	13.76	16.50	56.84	0.13%	0.16%	0.17%	0.17%	0.21%		
	70%	10.94	11.64	13.69	20.73	26.23	83.23	0.13%	0.15%	0.22%	0.22%	0.27%		
	80%	14.30	16.08	19.81	33.58	47.06	130.82	0.15%	0.21%	0.25%	0.32%	0.39%		
	90%	18.75	22.88	30.37	63.47	116.73	252.19	0.16%	0.24%	0.24%	0.40%	0.68%		
	95%	21.57	27.80	39.18	99.87	273.60	462.01	0.16%	0.25%	0.29%	0.52%	1.13%		
QNA	10%	0.89	1.19	1.41	1.78	2.05	7.33	-0.16%	43.68%	58.46%	60.92%	67.37%	48.21%	
	20%	1.92	2.58	3.05	3.85	4.41	15.81	0.12%	42.59%	56.15%	56.58%	60.63%	45.24%	
	30%	3.13	4.19	4.92	6.20	7.00	25.45	0.03%	40.80%	51.84%	48.97%	49.71%	39.93%	
	40%	4.55	6.09	7.07	8.85	9.83	36.38	-0.07%	38.27%	45.83%	39.00%	35.45%	32.69%	
	50%	6.25	8.34	9.54	11.88	13.01	49.03	-0.08%	34.65%	37.75%	26.74%	19.18%	23.62%	
	60%	8.33	11.07	12.46	15.55	17.00	64.41	0.03%	30.07%	28.04%	12.97%	3.04%	13.33%	
	70%	10.94	14.47	16.06	20.46	23.03	84.96	-0.03%	24.34%	17.33%	-1.30%	-12.20%	10.43%	
	80%	14.29	18.91	20.87	28.31	34.94	117.31	-0.10%	17.61%	5.35%	-15.70%	-25.75%	16.28%	
	90%	18.75	25.08	28.18	44.84	71.87	188.72	0.00%	9.66%	-7.20%	-29.35%	-38.43%	26.92%	
	95%	21.59	29.26	33.77	63.39	147.30	295.31	0.09%	5.28%	-13.81%	-36.53%	-46.16%	36.73%	28.36%
QNET	10%	0.89	0.75	0.77	0.91	1.02	4.34	-0.16%	-9.61%	-13.82%	-17.78%	-16.50%	12.20%	
	20%	1.92	1.63	1.68	2.03	2.31	9.57	0.12%	-9.61%	-13.82%	-17.65%	-15.94%	12.10%	
	30%	3.13	2.69	2.79	3.42	3.96	15.99	0.03%	-9.62%	-14.00%	-17.78%	-15.37%	12.10%	
	40%	4.55	3.97	4.15	5.22	6.16	24.06	-0.06%	-9.71%	-14.32%	-18.04%	-15.05%	12.27%	
	50%	6.25	5.56	5.87	7.60	9.25	34.53	-0.08%	-10.16%	-15.22%	-18.92%	-15.31%	12.95%	
	60%	8.33	7.57	8.10	10.90	13.85	48.75	0.02%	-11.02%	-16.80%	-20.80%	-16.10%	14.24%	
	70%	10.94	10.17	11.07	15.72	21.37	69.27	-0.03%	-12.64%	-19.13%	-24.20%	-18.50%	16.78%	
	80%	14.29	13.61	15.22	23.21	35.63	101.96	-0.10%	-15.35%	-23.18%	-30.86%	-24.27%	22.06%	
	90%	18.75	18.35	21.57	36.04	70.78	165.49	0.00%	-19.78%	-28.97%	-43.21%	-39.37%	34.38%	
	95%	21.59	21.60	26.88	49.16	124.51	243.74	0.09%	-22.30%	-31.39%	-50.78%	-54.49%	47.25%	33.93%
IR Method with QNA	10%	0.89	0.95	0.84	0.64	0.46	3.79	-0.16%	14.71%	-5.25%	-41.81%	-62.24%	28.22%	
	20%	1.92	2.08	1.87	1.49	1.11	8.47	0.12%	15.13%	-4.04%	-39.58%	-59.66%	27.24%	
	30%	3.13	3.44	3.15	2.62	2.03	14.38	0.03%	15.66%	-2.73%	-37.06%	-56.54%	26.08%	
	40%	4.55	5.12	4.79	4.20	3.43	22.08	-0.07%	16.30%	-1.19%	-33.98%	-52.79%	24.70%	
	50%	6.25	7.23	6.94	6.52	5.64	32.59	-0.08%	16.78%	0.26%	-30.46%	-48.31%	23.17%	
	60%	8.33	9.98	9.90	10.13	9.48	47.82	0.03%	17.21%	1.74%	-26.39%	-42.56%	21.63%	
	70%	10.94	13.67	14.18	16.28	16.97	72.04	-0.03%	17.43%	3.60%	-21.45%	-35.32%	19.51%	
	80%	14.29	18.91	20.87	28.31	34.94	117.31	-0.10%	17.61%	5.35%	-15.70%	-25.75%	16.28%	
	90%	18.75	26.89	32.60	58.20	102.39	238.83	0.00%	17.54%	7.37%	-8.31%	-12.28%	10.25%	
	95%	21.59	32.66	42.41	95.30	262.89	454.85	0.09%	17.49%	8.26%	-4.58%	-3.91%	5.06%	11.43%
Single-Point (80%) IR Method w/ Historical Data	10%	0.89	0.78	0.79	0.91	0.96	4.34	-0.16%	-6.42%	-10.97%	-17.46%	-21.67%	12.36%	
	20%	1.92	1.70	1.76	2.08	2.20	9.67	0.12%	-5.75%	-9.76%	-15.69%	-19.64%	11.22%	
	30%	3.13	2.83	2.97	3.59	3.87	16.39	0.03%	-4.92%	-8.46%	-13.79%	-17.22%	9.91%	
	40%	4.55	4.23	4.51	5.64	6.20	25.13	-0.07%	-3.94%	-6.91%	-11.46%	-14.52%	8.37%	
	50%	6.25	6.01	6.55	8.53	9.65	36.99	-0.08%	-3.00%	-5.42%	-8.97%	-11.62%	6.75%	
	60%	8.33	8.34	9.35	12.89	15.17	54.09	0.03%	-1.99%	-3.87%	-6.32%	-8.06%	4.83%	
	70%	10.94	11.53	13.43	20.05	25.14	81.07	-0.03%	-0.99%	-1.90%	-3.31%	-4.17%	2.59%	
	80%	14.30	16.10	19.82	33.48	47.15	130.85	0.00%	0.15%	0.04%	-0.29%	0.19%	0.17%	
	90%	18.75	23.21	31.10	65.55	122.63	261.23	0.00%	1.47%	2.40%	3.28%	5.05%	3.58%	
	95%	21.59	28.42	40.57	104.27	291.41	486.26	0.09%	2.26%	3.55%	4.40%	6.51%	5.25%	4.33%
Two-Point (70/80%) IR Method w/ Historical Data	10%	0.89	0.84	0.94	1.30	1.51	5.48	-0.16%	0.86%	5.22%	17.26%	23.49%	10.80%	
	20%	1.92	1.81	2.03	2.79	3.23	11.78	0.12%	0.41%	3.93%	13.30%	17.64%	8.25%	
	30%	3.13	2.98	3.33	4.56	5.27	19.27	0.03%	0.14%	2.78%	9.60%	12.61%	5.96%	
	40%	4.55	4.40	4.94	6.78	7.85	28.53	-0.07%	0.05%	1.92%	6.57%	8.23%	4.06%	
	50%	6.25	6.19	7.00	9.74	11.40	40.58	-0.08%	-0.08%	1.02%	3.90%	4.44%	2.35%	
	60%	8.33	8.50	9.76	14.00	16.81	57.40	0.03%	-0.10%	0.28%	1.70%	1.87%	1.02%	
	70%	10.94	11.63	13.70	20.80	26.30	83.37	0.00%	-0.09%	0.08%	0.33%	0.27%	0.20%	
	80%	14.30	16.10	19.82	33.48	47.15	130.85	0.00%	0.15%	0.04%	-0.29%	0.19%	0.17%	
	90%	18.75	23.03	30.59	64.02	119.68	256.07	0.00%	0.68%	0.73%	0.87%	2.53%	1.54%	
	95%	21.59	28.11	39.66	101.42	285.04	475.81	0.09%	1.14%	1.22%	1.54%	4.18%	2.99%	2.14%

Different from the previous two cases, the bottleneck is the fifth server, rather than the third one. The bottleneck is further away from the initial renewal arrival process. Although all the percentage errors become larger, the three IR methods still outperform QNA and QNET, especially in heavy traffic. Furthermore, in both QNA and QNET, the percentage errors seem to increase at the downstream servers, which might indicate that the total average errors could become larger when more servers are added to the sequence.

Case B-3 possesses the nearly-linear relationship of the intrinsic ratios as shown in Figure 6.6, which explains why IR methods can perform well. However, in Figure 6.6, the intrinsic ratios of the forth and fifth servers decrease faster in heavy traffic, which explains why the IR methods with historical data give larger percentage errors for the forth and fifth servers than the percentage errors of the second and the third servers in heavy traffic.

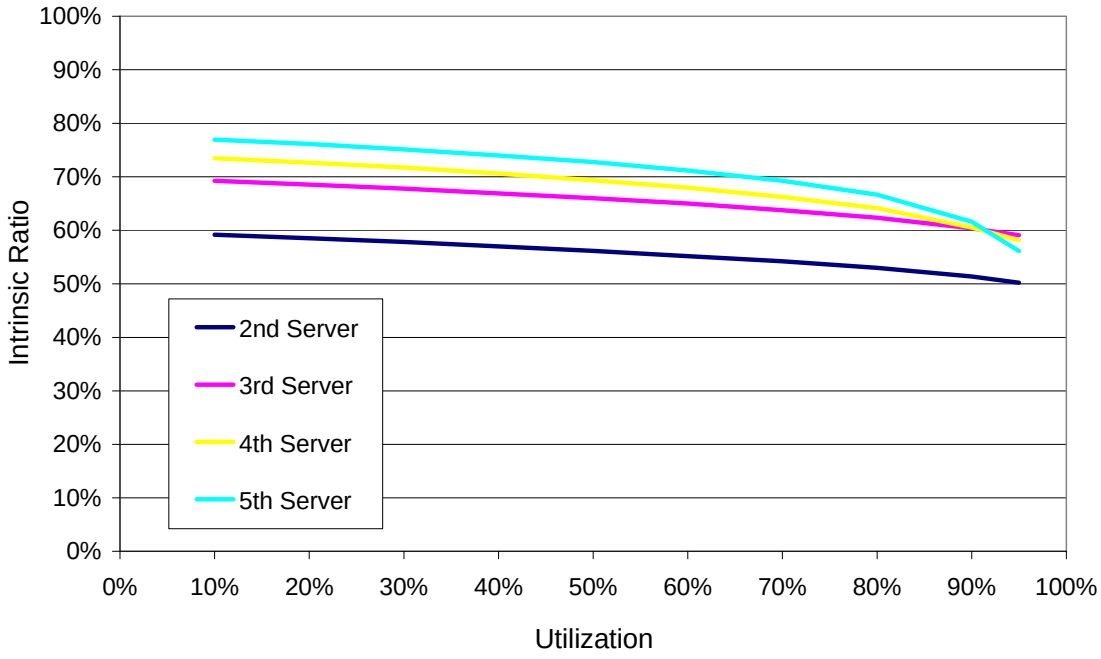

Figure 6.6 Intrinsic Ratios of Case B-3

Case B-4:

- Poisson arrivals

- Service time mean: (30, 28, 25, 23, 20) and SCV: (0.25, 0.25, 0.25, 0.25, 0.25)

Table 6.12 Queueing time approximations in Case B-4

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	2.09	1.08	0.71	0.55	0.37	4.80	0.13%	0.20%	0.19%	0.21%	0.19%		
	20%	4.69	2.38	1.54	1.18	0.78	10.57	0.12%	0.14%	0.15%	0.13%	0.16%		
	30%	8.03	3.98	2.51	1.91	1.23	17.67	0.09%	0.13%	0.13%	0.14%	0.12%		
	40%	12.49	6.02	3.68	2.76	1.75	26.70	0.12%	0.14%	0.14%	0.12%	0.13%		
	50%	18.75	8.71	5.13	3.78	2.34	38.71	0.12%	0.13%	0.13%	0.13%	0.13%		
	60%	28.15	12.43	7.00	5.03	3.04	55.64	0.13%	0.15%	0.12%	0.12%	0.12%		
	70%	43.77	17.98	9.47	6.62	3.86	81.70	0.16%	0.17%	0.12%	0.12%	0.12%		
	80%	74.99	27.15	12.99	8.72	4.87	128.73	0.21%	0.19%	0.15%	0.13%	0.13%		
	90%	168.36	45.10	18.34	11.66	6.17	249.63	0.41%	0.29%	0.19%	0.14%	0.13%		
	95%	356.27	62.06	22.21	13.58	6.96	461.07	0.93%	0.34%	0.20%	0.15%	0.13%		
QNA	10%	2.08	1.79	1.40	1.18	0.88	7.33	-0.09%	65.81%	97.08%	111.85%	136.42%	52.69%	
	20%	4.69	3.92	2.99	2.45	1.78	15.83	0.01%	64.53%	94.22%	106.83%	128.85%	49.72%	
	30%	8.04	6.44	4.70	3.74	2.65	25.57	0.02%	61.59%	87.76%	96.06%	114.45%	44.72%	
	40%	12.50	9.42	6.51	4.99	3.41	36.84	0.10%	56.51%	76.94%	81.03%	94.68%	37.98%	
	50%	18.75	13.02	8.39	6.17	4.05	50.38	0.00%	49.46%	63.51%	63.50%	72.95%	30.16%	
	60%	28.13	17.46	10.37	7.32	4.63	67.91	-0.09%	40.52%	48.20%	45.37%	52.55%	22.12%	
	70%	43.75	23.28	12.59	8.59	5.28	93.48	-0.05%	29.47%	32.96%	29.70%	36.78%	14.48%	
	80%	75.00	31.77	15.49	10.33	6.19	138.78	0.01%	17.02%	19.21%	18.42%	26.93%	7.80%	
	90%	168.75	47.22	20.32	13.27	7.64	257.21	0.23%	4.72%	10.81%	13.78%	23.90%	3.04%	
	95%	356.25	62.77	24.49	15.60	8.68	467.80	-0.01%	1.16%	10.27%	14.87%	24.79%	1.47%	8.00%
QNET	10%	2.08	1.08	0.75	0.57	0.40	4.89	-0.09%	-0.07%	5.28%	3.37%	7.61%	1.81%	
	20%	4.69	2.40	1.64	1.24	0.85	10.83	0.01%	0.67%	6.81%	5.13%	9.58%	2.42%	
	30%	8.04	4.05	2.72	2.04	1.38	18.22	0.02%	1.60%	8.62%	6.85%	11.78%	3.15%	
	40%	12.50	6.17	4.05	3.00	1.99	27.71	0.10%	2.44%	9.98%	8.62%	13.80%	3.77%	
	50%	18.75	9.00	5.72	4.17	2.71	40.35	0.00%	3.31%	11.43%	10.37%	15.80%	4.23%	
	60%	28.13	12.96	7.88	5.63	3.58	58.17	-0.09%	4.27%	12.62%	11.83%	17.77%	4.62%	
	70%	43.75	18.90	10.76	7.50	4.63	85.55	-0.05%	5.09%	13.72%	13.35%	19.99%	4.76%	
	80%	75.00	28.79	14.78	9.98	5.94	134.50	0.01%	6.05%	13.76%	14.43%	21.91%	4.48%	
	90%	168.75	48.27	20.69	13.43	7.64	258.78	0.23%	7.04%	12.82%	15.17%	23.85%	3.67%	
	95%	356.25	66.36	24.93	15.75	8.70	471.99	-0.01%	6.93%	12.22%	16.04%	25.08%	2.38%	3.34%
IR Method with QNA	10%	2.08	0.72	0.57	0.48	0.36	4.21	-0.09%	-33.08%	-20.25%	-13.94%	-3.62%	12.37%	
	20%	4.69	1.61	1.25	1.04	0.77	9.36	0.01%	-32.37%	-18.70%	-12.02%	-1.33%	11.46%	
	30%	8.04	2.73	2.08	1.72	1.25	15.82	0.02%	-31.47%	-16.81%	-10.02%	1.31%	10.66%	
	40%	12.50	4.18	3.13	2.54	1.82	24.17	0.10%	-30.54%	-15.08%	-7.81%	3.93%	10.08%	
	50%	18.75	6.14	4.47	3.57	2.50	35.43	0.00%	-29.46%	-12.99%	-5.33%	6.74%	9.28%	
	60%	28.13	8.94	6.25	4.90	3.33	51.55	-0.09%	-28.09%	-10.63%	-2.70%	9.78%	8.43%	
	70%	43.75	13.23	8.75	6.66	4.38	76.77	-0.05%	-26.42%	-7.53%	0.61%	13.42%	7.40%	
	80%	75.00	20.69	12.50	9.12	5.71	123.03	0.01%	-23.79%	-3.74%	4.55%	17.23%	6.36%	
	90%	168.75	36.86	18.76	12.80	7.50	244.67	0.23%	-18.27%	2.27%	9.77%	21.57%	4.61%	
	95%	356.25	54.93	23.76	15.42	8.64	458.99	-0.01%	-11.49%	6.97%	13.56%	24.12%	2.65%	4.89%
Single-Point (80%) IR Method w/ Historical Data	10%	2.08	0.95	0.59	0.46	0.30	4.38	-0.09%	-12.20%	-17.18%	-17.75%	-17.83%	8.76%	
	20%	4.69	2.11	1.30	1.00	0.66	9.75	0.01%	-11.27%	-15.58%	-15.92%	-15.87%	7.76%	
	30%	8.04	3.58	2.16	1.64	1.07	16.49	0.02%	-10.09%	-13.62%	-14.01%	-13.62%	6.68%	
	40%	12.50	5.49	3.25	2.43	1.55	25.22	0.10%	-8.87%	-11.82%	-11.89%	-11.39%	5.65%	
	50%	18.75	8.06	4.64	3.42	2.13	37.00	0.00%	-7.45%	-9.65%	-9.52%	-8.99%	4.43%	
	60%	28.13	11.72	6.49	4.68	2.84	53.87	-0.09%	-5.66%	-7.20%	-7.01%	-6.40%	3.20%	
	70%	43.75	17.36	9.09	6.37	3.73	80.30	-0.05%	-3.47%	-3.98%	-3.85%	-3.29%	1.72%	
	80%	74.99	27.15	12.99	8.72	4.87	128.72	0.00%	-0.01%	-0.04%	-0.08%	-0.05%	0.01%	
	90%	168.75	48.36	19.48	12.23	6.40	255.22	0.23%	7.24%	6.20%	4.91%	3.66%	2.24%	
	95%	356.25	72.07	24.67	14.73	7.36	475.09	-0.01%	16.13%	11.08%	8.53%	5.83%	3.05%	2.65%
Two-Point (70/80%) IR Method w/ Historical Data	10%	2.08	1.19	0.78	0.59	0.38	5.02	-0.09%	10.21%	8.78%	5.70%	2.71%	4.50%	
	20%	4.69	2.58	1.65	1.24	0.80	10.95	0.01%	8.14%	7.11%	4.62%	2.15%	3.55%	
	30%	8.04	4.24	2.65	1.98	1.26	18.15	0.02%	6.30%	5.73%	3.50%	1.79%	2.74%	
	40%	12.50	6.29	3.83	2.83	1.77	27.21	0.10%	4.42%	3.98%	2.46%	1.27%	1.93%	
	50%	18.75	8.94	5.26	3.83	2.36	39.15	0.00%	2.67%	2.49%	1.53%	0.76%	1.13%	
	60%	28.13	12.58	7.07	5.06	3.05	55.89	-0.09%	1.22%	1.11%	0.57%	0.28%	0.52%	
	70%	43.77	17.99	9.50	6.63	3.86	81.75	0.00%	0.05%	0.32%	0.07%	0.16%	0.06%	
	80%	74.99	27.15	12.99	8.72	4.87	128.72	0.00%	-0.01%	-0.04%	-0.08%	-0.05%	0.01%	
	90%	168.75	46.60	18.61	11.73	6.17	251.85	0.23%	3.33%	1.45%	0.64%	-0.04%	0.90%	
	95%	356.25	68.13	23.01	13.83	6.97	468.19	-0.01%	9.78%	3.62%	1.90%	0.16%	1.55%	1.10%

Since the bottleneck is the first server and the arrival process is Poisson, the longest queueing time can be calculated exactly by the P-K formula. Therefore, the approximation errors are smaller than the previous cases. The two-point IR methods with historical data perform the best.

The interchangeability theorem, proved by Weber (1979), says that for tandem queues with exponential service times, the output process has the same distribution no matter what the sequence of the servers is. This implies, for any ordering of the servers, that the system queueing times are the same. The ordering also does not matter when all service time are deterministic, which is proved by Friedman (1965). Therefore, Suresh and Whitt (1990a) conjectured that the system queueing times should be similar if all service time SCV are the same and between 0 and 1, but just with different ordering. This phenomenon can be observed from the results of Case B-3 and Case B-4. Furthermore, the nearly-linear relationship of the intrinsic ratios of Case B-4 is shown in Figure 6.7.

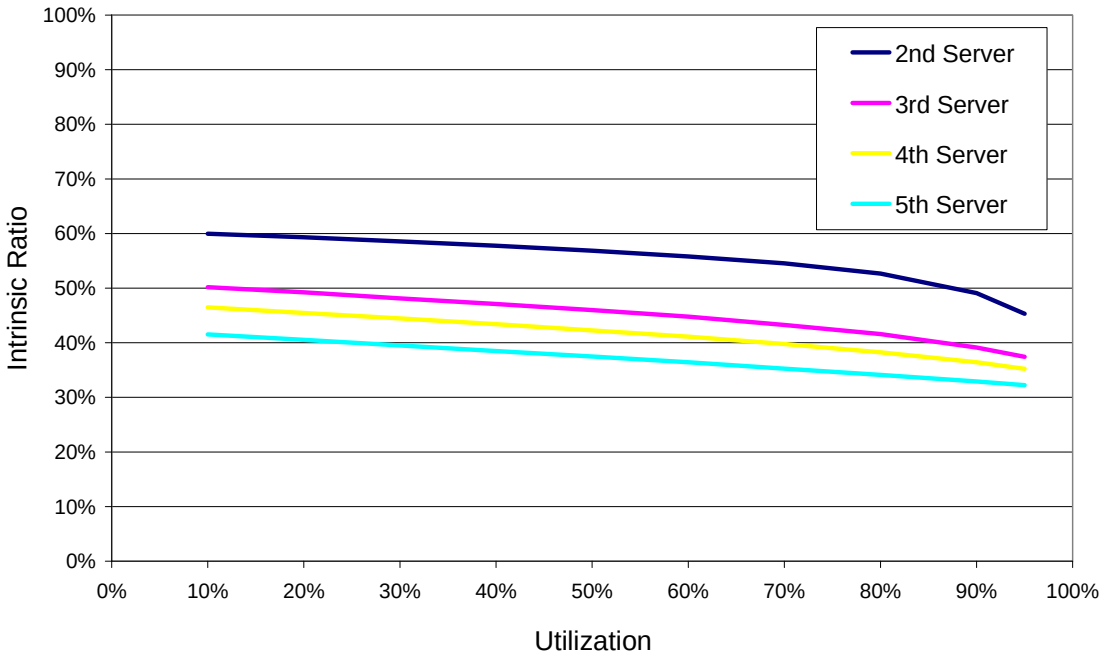

Figure 6.7 Intrinsic Ratios of Case B-4

Case B-5:

- Poisson arrivals

- Service time mean: (30, 25, 28, 20, 30) and SCV: (0.25, 0.25, 0.25, 0.25, 0.25)

Table 6.13 Queueing time approximations in Case B-5

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	2.08	0.78	1.05	0.35	1.24	5.50	0.18%	0.25%	0.22%	0.26%	0.20%		
	20%	4.69	1.68	2.31	0.73	2.77	12.18	0.16%	0.19%	0.19%	0.22%	0.16%		
	30%	8.02	2.74	3.86	1.15	4.68	20.46	0.12%	0.16%	0.17%	0.19%	0.15%		
	40%	12.49	4.03	5.83	1.63	7.19	31.18	0.13%	0.16%	0.16%	0.16%	0.16%		
	50%	18.76	5.61	8.45	2.18	10.70	45.70	0.15%	0.16%	0.15%	0.15%	0.16%		
	60%	28.09	7.61	12.12	2.82	15.89	66.53	0.18%	0.16%	0.19%	0.17%	0.19%		
	70%	43.74	10.27	17.63	3.59	24.65	99.88	0.23%	0.19%	0.23%	0.18%	0.27%		
	80%	75.22	13.87	26.88	4.52	42.53	163.01	0.34%	0.19%	0.26%	0.16%	0.33%		
	90%	168.82	19.10	45.13	5.71	97.05	335.82	0.55%	0.22%	0.33%	0.14%	0.62%		
	95%	355.23	22.63	62.77	6.45	208.44	655.51	0.92%	0.21%	0.40%	0.16%	0.99%		
QNA	10%	2.08	1.41	1.78	0.88	2.05	8.20	0.12%	81.75%	69.99%	153.82%	64.41%	49.23%	
	20%	4.69	3.05	3.86	1.81	4.36	17.77	-0.07%	81.42%	66.80%	148.11%	57.72%	45.92%	
	30%	8.04	4.93	6.21	2.72	6.85	28.75	0.15%	79.52%	60.75%	136.36%	46.50%	40.51%	
	40%	12.50	7.06	8.84	3.57	9.48	41.45	0.06%	75.35%	51.59%	118.79%	31.72%	32.95%	
	50%	18.75	9.49	11.82	4.32	12.35	56.72	-0.07%	68.98%	39.85%	98.27%	15.44%	24.17%	
	60%	28.13	12.25	15.32	4.98	15.92	76.60	0.13%	60.94%	26.45%	76.68%	0.17%	15.13%	
	70%	43.75	15.44	19.85	5.64	21.46	106.14	0.03%	50.36%	12.58%	57.24%	-12.94%	12.66%	
	80%	75.00	19.25	26.82	6.47	32.85	160.39	-0.29%	38.81%	-0.23%	43.26%	-22.75%	10.61%	
	90%	168.75	24.09	41.33	7.78	69.09	311.04	-0.04%	26.13%	-8.42%	36.12%	-28.82%	11.58%	
	95%	356.25	27.22	57.75	8.74	143.50	593.46	0.29%	20.31%	-7.99%	35.48%	-31.16%	11.88%	13.56%
QNET	10%	2.08	0.82	0.99	0.40	0.99	5.28	0.12%	5.70%	-6.01%	16.86%	-20.64%	7.73%	
	20%	4.69	1.79	2.19	0.86	2.22	11.76	-0.07%	6.75%	-5.10%	18.62%	-19.89%	7.56%	
	30%	8.04	2.97	3.71	1.39	3.79	19.89	0.15%	8.10%	-4.01%	20.76%	-19.06%	7.43%	
	40%	12.50	4.40	5.65	2.00	5.86	30.42	0.06%	9.37%	-3.11%	22.70%	-18.48%	7.27%	
	50%	18.75	6.21	8.24	2.72	8.75	44.67	-0.07%	10.60%	-2.46%	24.86%	-18.17%	7.23%	
	60%	28.13	8.54	11.86	3.58	13.05	65.15	0.13%	12.13%	-2.10%	26.83%	-17.87%	7.23%	
	70%	43.75	11.63	17.26	4.62	20.19	97.45	0.03%	13.24%	-2.11%	28.83%	-18.11%	7.25%	
	80%	75.00	15.89	26.15	5.94	34.46	157.44	-0.29%	14.61%	-2.74%	31.37%	-18.96%	7.64%	
	90%	168.75	21.86	43.57	7.64	77.16	318.99	-0.04%	14.45%	-3.46%	33.81%	-20.49%	7.80%	
	95%	356.25	25.70	60.97	8.71	159.23	610.86	0.29%	13.58%	-2.87%	35.09%	-23.61%	8.75%	8.12%
IR Method with QNA	10%	2.08	0.57	0.72	0.36	0.91	4.64	0.12%	-26.64%	-31.24%	3.11%	-26.68%	16.01%	
	20%	4.69	1.25	1.61	0.77	2.05	10.37	-0.07%	-25.42%	-30.44%	5.54%	-25.79%	15.50%	
	30%	8.04	2.09	2.72	1.25	3.52	17.62	0.15%	-23.86%	-29.42%	8.47%	-24.76%	14.95%	
	40%	12.50	3.13	4.17	1.82	5.47	27.10	0.06%	-22.18%	-28.43%	11.42%	-23.90%	14.32%	
	50%	18.75	4.48	6.13	2.50	8.21	40.07	-0.07%	-20.24%	-27.46%	14.81%	-23.22%	13.74%	
	60%	28.13	6.27	8.92	3.33	12.32	58.96	0.13%	-17.64%	-26.41%	18.26%	-22.49%	13.03%	
	70%	43.75	8.78	13.20	4.38	19.16	89.27	0.03%	-14.56%	-25.11%	21.97%	-22.26%	12.23%	
	80%	75.00	12.54	20.65	5.71	32.85	146.75	-0.29%	-9.59%	-23.18%	26.48%	-22.75%	11.44%	
	90%	168.75	18.81	36.78	7.50	73.91	305.75	-0.04%	-1.55%	-18.50%	31.31%	-23.84%	10.02%	
	95%	356.25	23.82	54.82	8.64	156.04	599.56	0.29%	5.27%	-12.67%	33.94%	-25.14%	9.88%	10.76%
Single-Point (80%) IR Method w/ Historical Data	10%	2.08	0.63	0.94	0.28	1.18	5.11	0.12%	-18.95%	-10.48%	-18.45%	-5.47%	7.12%	
	20%	4.69	1.39	2.09	0.61	2.65	11.42	-0.07%	-17.61%	-9.43%	-16.53%	-4.32%	6.22%	
	30%	8.04	2.31	3.55	0.99	4.54	19.42	0.15%	-15.89%	-8.11%	-14.21%	-3.00%	5.21%	
	40%	12.50	3.46	5.43	1.44	7.06	29.89	0.06%	-14.02%	-6.82%	-11.87%	-1.88%	4.17%	
	50%	18.75	4.95	7.98	1.98	10.59	44.24	-0.07%	-11.89%	-5.55%	-9.20%	-1.01%	3.19%	
	60%	28.13	6.93	11.61	2.64	15.88	65.18	0.13%	-9.01%	-4.19%	-6.47%	-0.07%	2.14%	
	70%	43.75	9.70	17.19	3.46	24.71	98.80	0.03%	-5.60%	-2.50%	-3.53%	0.23%	1.21%	
	80%	75.22	13.85	26.89	4.52	42.35	162.83	0.00%	-0.12%	0.01%	0.04%	-0.41%	0.12%	
	90%	168.75	20.78	47.89	5.93	95.29	338.65	-0.04%	8.76%	6.11%	3.86%	-1.81%	1.93%	
	95%	356.25	26.32	71.37	6.83	201.18	661.94	0.29%	16.30%	13.70%	5.93%	-3.49%	3.20%	2.45%
Two-Point (70/80%) IR Method w/ Historical Data	10%	2.08	0.88	1.11	0.35	1.16	5.58	0.12%	13.89%	5.57%	0.93%	-7.02%	4.72%	
	20%	4.69	1.87	2.42	0.73	2.61	12.31	-0.07%	11.01%	4.49%	0.47%	-5.67%	3.72%	
	30%	8.04	2.98	4.00	1.16	4.48	20.66	0.15%	8.46%	3.66%	0.35%	-4.14%	2.85%	
	40%	12.50	4.26	5.99	1.63	6.99	31.38	0.06%	5.89%	2.73%	0.09%	-2.81%	1.95%	
	50%	18.75	5.81	8.60	2.18	10.51	45.84	-0.07%	3.42%	1.70%	0.05%	-1.71%	1.17%	
	60%	28.13	7.73	12.20	2.82	15.81	66.68	0.13%	1.52%	0.72%	-0.12%	-0.54%	0.49%	
	70%	43.74	10.26	17.63	3.58	24.65	99.85	0.00%	-0.14%	0.00%	-0.25%	0.00%	0.02%	
	80%	75.22	13.85	26.89	4.52	42.35	162.83	0.00%	-0.12%	0.01%	0.04%	-0.41%	0.12%	
	90%	168.75	19.57	46.66	5.73	95.52	336.24	-0.04%	2.47%	3.39%	0.33%	-1.58%	1.08%	
	95%	356.25	24.03	68.63	6.48	201.89	657.28	0.29%	6.20%	9.33%	0.54%	-3.15%	2.27%	1.50%

When bottlenecks are the first and the fifth servers, the average percentage errors become larger than the errors in Case B-4. Since there are two bottlenecks and the utilization of the third server is also close to the bottlenecks (i.e. 28 vs. 30), QNET performs relatively well. The IR methods with historical data still perform the best.

Case B-5 possesses the nearly-linear relationship of the intrinsic ratios as shown in Figure 6.8, which explains why the IR methods can perform well. It should be noted that the intrinsic ratio of the fifth server (one of the two bottlenecks) is almost flat, and QNA performs very well at 60% utilization (with 0.17% error). If we can approximate the intrinsic ratio by QNA at 60% utilization, the total average errors will become smaller (4.38%). However, following the general rules, we approximate the intrinsic ratio by QNA at 80% utilization, which induces the larger errors (10.76%). It tells us that there is still much room for improving the errors of the IR method by better understanding the structure of QNA errors.

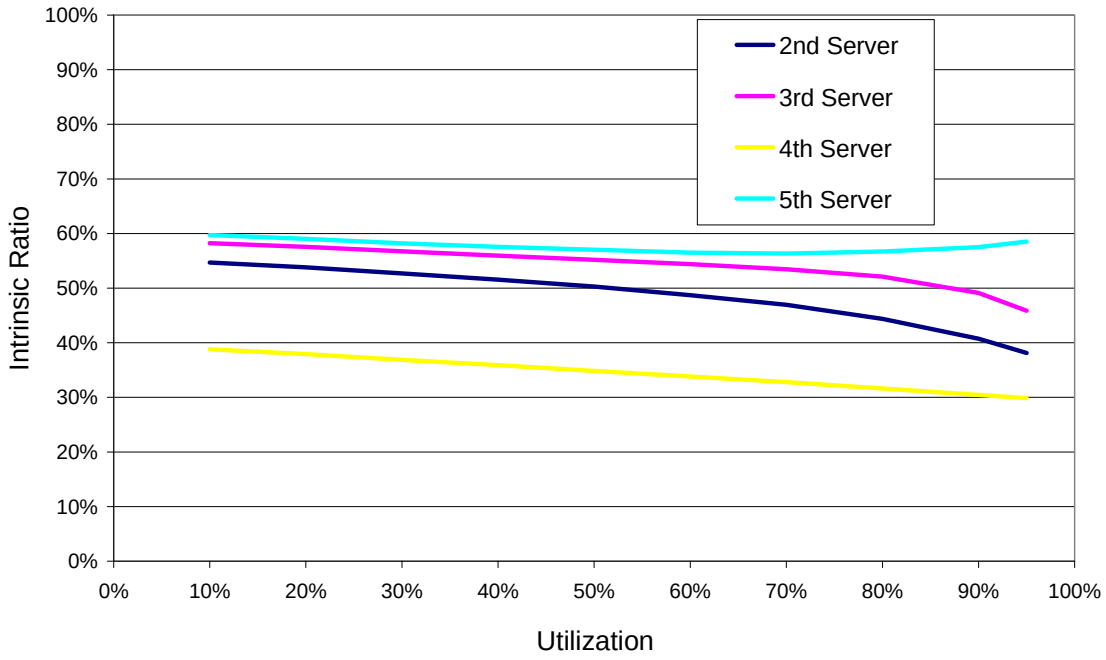

Figure 6.8 Intrinsic Ratios of Case B-5

Case B-6:

- Poisson arrivals

- Service time mean: (23, 25, 28, 20, 30) and SCV: (8, 0.25, 0.8, 0.5, 1)

Table 6.14 Queueing time approximations in Case B-6

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	8.59	2.95	2.73	1.10	3.47	18.84	0.34%	0.18%	0.21%	0.20%	0.19%		
	20%	18.73	7.34	6.44	2.43	8.14	43.09	0.34%	0.20%	0.19%	0.20%	0.18%		
	30%	30.84	13.60	11.51	4.04	14.50	74.50	0.30%	0.18%	0.19%	0.18%	0.17%		
	40%	45.76	22.52	18.62	6.00	23.45	116.35	0.36%	0.20%	0.20%	0.17%	0.21%		
	50%	64.12	34.90	28.55	8.28	36.29	172.14	0.36%	0.21%	0.24%	0.18%	0.25%		
	60%	88.47	52.94	43.46	11.00	56.02	251.89	0.36%	0.25%	0.30%	0.21%	0.31%		
	70%	119.46	78.84	66.53	14.16	88.85	367.84	0.43%	0.25%	0.33%	0.18%	0.37%		
	80%	163.95	118.78	107.67	17.86	154.09	562.36	0.54%	0.31%	0.43%	0.19%	0.45%		
	90%	230.49	183.33	194.44	22.12	336.76	967.14	0.63%	0.37%	0.53%	0.19%	0.81%		
	95%	276.36	234.41	292.42	24.62	680.01	1507.81	0.59%	0.36%	0.78%	0.19%	1.51%		
QNA	10%	8.59	1.47	2.65	1.10	3.39	17.19	0.05%	-50.34%	-3.06%	-0.12%	-2.51%	8.83%	
	20%	18.74	3.54	6.23	2.50	7.94	38.95	0.08%	-51.82%	-3.27%	2.89%	-2.54%	10.00%	
	30%	30.92	6.75	11.43	4.40	14.34	67.84	0.24%	-50.36%	-0.69%	8.89%	-1.10%	10.09%	
	40%	45.78	11.93	19.20	6.92	23.40	107.23	0.03%	-47.03%	3.11%	15.43%	-0.24%	10.46%	
	50%	64.34	20.34	30.87	10.10	36.09	161.74	0.34%	-41.71%	8.13%	21.90%	-0.53%	11.10%	
	60%	88.17	34.14	48.53	13.81	53.99	238.63	-0.34%	-35.52%	11.66%	25.49%	-3.62%	11.51%	
	70%	119.88	57.16	75.85	17.77	80.72	351.38	0.36%	-27.50%	14.01%	25.45%	-9.15%	11.74%	
	80%	164.17	97.08	120.88	21.58	128.14	531.86	0.13%	-18.27%	12.27%	20.82%	-16.84%	11.52%	
	90%	230.37	171.85	208.46	24.96	259.86	895.50	-0.05%	-6.26%	7.21%	12.81%	-22.83%	10.89%	
	95%	277.48	235.76	297.48	26.58	519.43	1356.73	0.41%	0.58%	1.73%	7.95%	-23.61%	11.28%	11.20%
QNET	10%	8.59	5.83	2.33	0.93	2.76	20.45	0.05%	97.38%	-14.54%	-14.93%	-20.66%	22.07%	
	20%	18.74	12.77	5.12	2.00	6.19	44.83	0.08%	74.00%	-20.49%	-17.67%	-23.96%	21.23%	
	30%	30.92	21.18	8.52	3.25	10.61	74.47	0.24%	55.73%	-26.02%	-19.74%	-26.85%	20.59%	
	40%	45.78	31.61	12.75	4.71	16.50	111.34	0.03%	40.37%	-31.52%	-21.53%	-29.67%	19.96%	
	50%	64.34	44.92	18.19	6.45	24.76	158.67	0.34%	28.71%	-36.27%	-22.09%	-31.76%	19.72%	
	60%	88.17	62.59	25.50	8.58	37.21	222.05	-0.34%	18.22%	-41.32%	-21.99%	-33.58%	19.51%	
	70%	119.88	87.26	36.02	11.24	58.04	312.45	0.36%	10.68%	-45.86%	-20.61%	-34.67%	19.87%	
	80%	164.17	124.21	53.14	14.67	99.78	455.98	0.13%	4.57%	-50.64%	-17.85%	-35.24%	20.92%	
	90%	230.37	185.37	89.01	19.27	224.88	748.89	-0.05%	1.11%	-54.22%	-12.92%	-33.22%	22.99%	
	95%	277.48	233.76	128.95	22.20	473.64	1136.02	0.41%	-0.28%	-55.90%	-9.80%	-30.35%	24.81%	22.59%
IR Method with QNA	10%	8.59	4.87	5.00	1.02	3.73	23.20	0.05%	64.72%	83.06%	-7.33%	7.30%	23.97%	
	20%	18.74	10.64	11.03	2.19	8.36	50.96	0.08%	44.99%	71.17%	-10.04%	2.65%	19.41%	
	30%	30.92	17.61	18.46	3.56	14.28	84.81	0.24%	29.45%	60.29%	-12.02%	-1.52%	15.74%	
	40%	45.78	26.17	27.85	5.17	22.12	127.09	0.03%	16.21%	49.57%	-13.74%	-5.70%	12.94%	
	50%	64.34	36.96	40.12	7.11	32.99	181.53	0.34%	5.89%	40.55%	-14.11%	-9.07%	10.63%	
	60%	88.17	50.98	56.90	9.49	49.14	254.67	-0.34%	-3.71%	30.93%	-13.79%	-12.27%	9.57%	
	70%	119.88	69.95	81.36	12.45	75.72	359.35	0.36%	-11.28%	22.29%	-12.10%	-14.78%	10.60%	
	80%	164.17	97.08	120.88	16.26	128.03	526.43	0.13%	-18.27%	12.27%	-8.96%	-16.91%	11.17%	
	90%	230.37	139.25	198.24	21.34	281.88	871.09	-0.05%	-24.05%	1.96%	-3.53%	-16.30%	10.72%	
	95%	277.48	170.64	277.10	24.58	585.12	1334.92	0.41%	-27.20%	-5.24%	-0.17%	-13.95%	11.62%	11.33%
Single-Point (80%) IR Method w/ Historical Data	10%	8.59	6.01	4.29	1.12	4.96	24.97	0.05%	103.38%	57.30%	1.78%	42.94%	32.55%	
	20%	18.74	13.13	9.49	2.40	11.07	54.84	0.08%	78.93%	47.34%	-1.19%	35.89%	27.41%	
	30%	30.92	21.71	15.92	3.91	18.76	91.22	0.24%	59.66%	38.27%	-3.37%	29.38%	22.81%	
	40%	45.78	32.25	24.09	5.68	28.78	136.59	0.03%	43.23%	29.37%	-5.26%	22.72%	17.93%	
	50%	64.34	45.51	34.83	7.81	42.42	194.91	0.34%	30.39%	22.00%	-5.66%	16.91%	13.77%	
	60%	88.17	62.70	49.62	10.42	62.18	273.08	-0.34%	18.42%	14.18%	-5.31%	11.00%	9.12%	
	70%	119.88	85.88	71.42	13.67	93.69	384.55	0.36%	8.93%	7.36%	-3.45%	5.45%	4.81%	
	80%	163.95	118.79	107.16	17.86	153.22	560.97	0.00%	0.00%	-0.48%	-0.01%	-0.57%	0.25%	
	90%	230.37	169.87	178.85	23.44	319.08	921.62	-0.05%	-7.34%	-8.02%	5.95%	-5.25%	4.98%	
	95%	277.48	207.52	253.56	26.99	632.36	1397.93	0.41%	-11.47%	-13.29%	9.65%	-7.01%	7.75%	7.02%
Two-Point (70/80%) IR Method w/ Historical Data	10%	8.59	2.46	2.35	1.41	3.15	17.95	0.05%	-16.71%	-14.05%	28.42%	-9.43%	8.07%	
	20%	18.74	6.50	5.72	2.94	7.52	41.43	0.08%	-11.45%	-11.26%	20.97%	-7.59%	6.29%	
	30%	30.92	12.60	10.55	4.64	13.69	72.39	0.24%	-7.38%	-8.40%	14.69%	-5.58%	4.63%	
	40%	45.78	21.45	17.50	6.53	22.52	113.79	0.03%	-4.73%	-6.01%	8.91%	-3.97%	3.15%	
	50%	64.34	34.13	27.65	8.69	35.54	170.34	0.34%	-2.23%	-3.16%	4.91%	-2.05%	1.77%	
	60%	88.17	52.30	42.84	11.20	55.61	250.11	-0.34%	-1.22%	-1.42%	1.77%	-0.73%	0.86%	
	70%	119.46	78.61	66.60	14.19	88.96	367.81	0.00%	-0.29%	0.11%	0.15%	0.13%	0.12%	
	80%	163.95	118.79	107.16	17.86	153.22	560.97	0.00%	0.00%	-0.48%	-0.01%	-0.57%	0.25%	
	90%	230.37	183.46	188.64	22.57	329.32	954.36	-0.05%	0.07%	-2.98%	1.99%	-2.21%	1.44%	
	95%	277.48	232.07	271.57	25.48	651.98	1458.59	0.41%	-1.00%	-7.13%	3.50%	-4.12%	3.53%	2.09%

In contrast to the previous case, the service time SCVs are not equal in Case B-6. The service time SCV of the first server is 8, which introduces large variability into the system. Since the slope of the intrinsic ratio of the second server is not close to zero as shown in Figure 6.9, there is a large percentage error at the second server in light traffic. Therefore, only the two-point IR method with historical data performs well in this case.

It should be noted that although QNA and QNET are heavy traffic approximations, both do not perform well in heavy traffic for this case. The percentage errors of QNA at the bottleneck (i.e. the fifth server) increase in heavy traffic. On the other hand, the percentage error of the two-point IR method is 3.53% at 95% utilization.

Except for the intrinsic ratio of the forth server in heavy traffic, all the other intrinsic ratios are greater than 1 in Figure 6.9. Even if there is only one server whose service time SCV is greater than the arrival process SCV, the ASIA system queueing time could still be a lower bound.

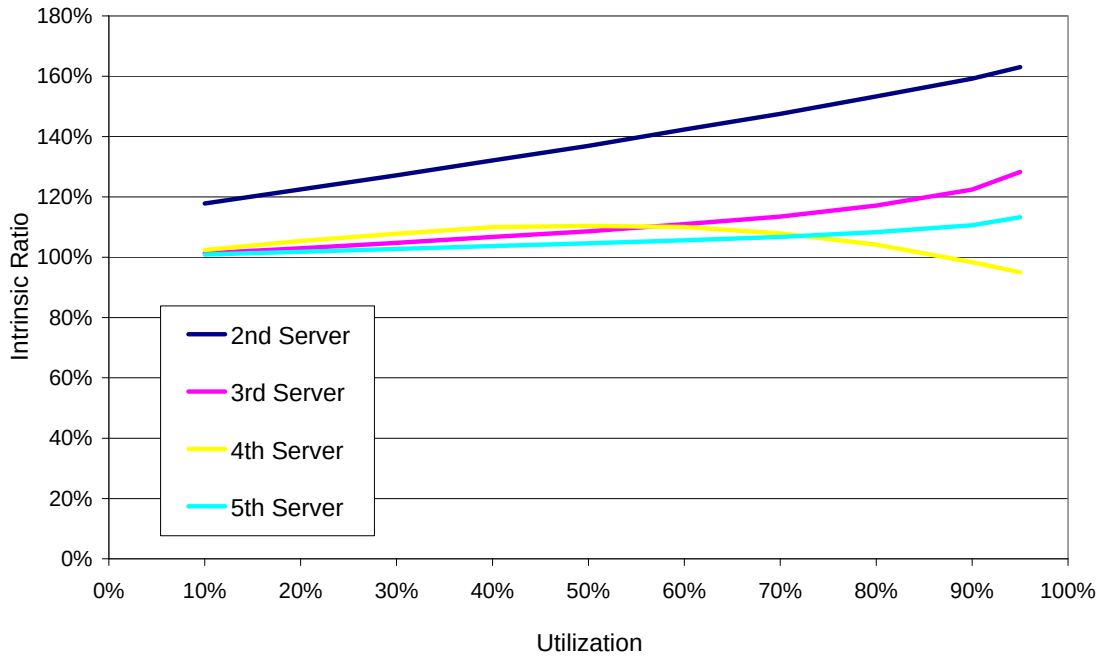

Figure 6.9 Intrinsic Ratios of Case B-6

Case B-7:

- Poisson arrivals

- Service time mean: (23, 25, 28, 20, 30) and SCV: (8, 2, 5, 4, 1)

Table 6.15 Queueing time approximations in Case B-7

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	8.61	4.79	9.77	5.09	5.84	34.10	0.24%	0.17%	0.18%	0.18%	0.13%		
	20%	18.75	11.34	22.39	11.67	14.19	78.34	0.26%	0.15%	0.19%	0.17%	0.12%		
	30%	30.87	20.30	38.91	20.09	25.96	136.13	0.21%	0.14%	0.19%	0.17%	0.12%		
	40%	45.83	32.46	61.02	30.72	42.52	212.55	0.26%	0.16%	0.22%	0.18%	0.12%		
	50%	64.29	49.10	91.23	43.94	66.29	314.84	0.27%	0.16%	0.22%	0.18%	0.14%		
	60%	87.90	72.17	134.47	60.57	101.95	457.06	0.28%	0.18%	0.24%	0.18%	0.17%		
	70%	119.24	105.37	199.62	81.56	159.78	665.57	0.34%	0.22%	0.34%	0.17%	0.19%		
	80%	164.36	156.19	311.03	109.01	270.80	1011.39	0.35%	0.21%	0.41%	0.23%	0.27%		
	90%	230.02	238.37	541.51	145.13	568.87	1723.90	0.39%	0.30%	0.61%	0.26%	0.49%		
	95%	277.43	303.39	785.62	168.53	1080.87	2615.83	0.47%	0.34%	0.77%	0.25%	0.95%		
QNA	10%	8.59	3.46	8.72	3.63	3.49	27.89	-0.15%	-27.81%	-10.79%	-28.66%	-40.24%	18.21%	
	20%	18.74	7.91	19.88	8.19	8.88	63.60	-0.03%	-30.26%	-11.20%	-29.83%	-37.42%	18.81%	
	30%	30.92	14.04	34.90	14.23	17.89	111.98	0.15%	-30.81%	-10.32%	-29.18%	-31.07%	17.81%	
	40%	45.78	22.86	55.85	22.39	32.88	179.76	-0.12%	-29.55%	-8.47%	-27.12%	-22.68%	15.43%	
	50%	64.34	35.97	86.04	33.36	57.29	277.00	0.08%	-26.74%	-5.69%	-24.07%	-13.57%	12.05%	
	60%	88.17	56.02	131.16	47.92	97.16	420.43	0.31%	-22.38%	-2.46%	-20.88%	-4.70%	8.13%	
	70%	119.88	87.78	202.38	67.07	165.82	642.93	0.54%	-16.70%	1.38%	-17.77%	3.78%	6.24%	
	80%	164.17	140.83	326.28	92.27	303.18	1026.74	-0.11%	-9.83%	4.90%	-15.35%	11.96%	7.90%	
	90%	230.37	237.48	589.51	126.26	711.05	1894.67	0.15%	-0.38%	8.86%	-13.00%	24.99%	12.20%	
	95%	277.48	318.88	877.63	148.12	1523.16	3145.27	0.02%	5.11%	11.71%	-12.11%	40.92%	21.80%	14.55%
QNET	10%	8.59	7.87	11.00	5.91	8.19	41.56	-0.15%	64.39%	12.57%	16.15%	40.16%	21.97%	
	20%	18.74	17.28	24.49	12.75	18.44	91.71	-0.03%	52.31%	9.40%	9.29%	29.96%	17.08%	
	30%	30.92	28.73	41.45	20.75	31.65	153.50	0.15%	41.54%	6.52%	3.30%	21.92%	12.76%	
	40%	45.78	42.97	63.39	30.25	49.30	231.69	-0.12%	32.39%	3.88%	-1.51%	15.94%	9.49%	
	50%	64.34	61.17	92.88	41.72	74.05	334.16	0.08%	24.59%	1.81%	-5.05%	11.71%	7.54%	
	60%	88.17	85.26	134.66	55.83	111.17	475.09	0.31%	18.14%	0.14%	-7.83%	9.04%	6.02%	
	70%	119.88	118.67	198.41	73.67	172.69	683.31	0.54%	12.62%	-0.61%	-9.67%	8.08%	5.40%	
	80%	164.17	168.20	307.70	96.95	293.34	1030.37	-0.12%	7.69%	-1.07%	-11.06%	8.32%	4.96%	
	90%	230.37	249.66	539.24	128.83	633.77	1781.87	0.15%	4.74%	-0.42%	-11.23%	11.41%	5.52%	
	95%	277.48	314.31	791.25	149.94	1248.64	2781.62	0.02%	3.60%	0.72%	-11.03%	15.52%	7.76%	6.81%
IR Method with QNA	10%	8.59	6.86	12.63	6.24	12.02	46.34	-0.15%	43.20%	29.31%	22.58%	105.74%	35.99%	
	20%	18.74	15.02	27.99	13.43	26.53	101.72	-0.03%	32.37%	25.01%	15.17%	86.98%	29.86%	
	30%	30.92	24.90	47.07	21.83	44.43	169.14	0.15%	22.67%	20.96%	8.67%	71.15%	24.25%	
	40%	45.78	37.11	71.43	31.76	67.11	253.18	-0.12%	14.33%	17.06%	3.38%	57.81%	19.17%	
	50%	64.34	52.58	103.67	43.66	96.90	361.16	0.08%	7.10%	13.63%	-0.62%	46.18%	14.88%	
	60%	88.17	72.85	148.46	58.22	138.09	505.78	0.31%	0.95%	10.40%	-3.89%	35.44%	11.69%	
	70%	119.88	100.57	215.19	76.41	199.63	711.69	0.54%	-4.56%	7.80%	-6.31%	24.94%	9.92%	
	80%	164.17	140.83	326.28	99.80	305.08	1036.17	-0.11%	-9.83%	4.90%	-8.45%	12.66%	7.35%	
	90%	230.37	204.88	553.31	130.99	553.43	1672.98	0.15%	-14.05%	2.18%	-9.74%	-2.71%	4.36%	
	95%	277.48	253.77	794.26	150.84	943.76	2420.10	0.02%	-16.36%	1.10%	-10.50%	-12.69%	8.15%	8.81%
Single-Point (80%) IR Method w/ Historical Data	10%	8.59	7.67	11.85	6.82	10.44	45.37	-0.15%	60.32%	21.26%	33.96%	78.61%	33.14%	
	20%	18.74	16.80	26.27	14.68	23.06	99.56	-0.03%	48.13%	17.33%	25.87%	62.47%	27.10%	
	30%	30.92	27.84	44.22	23.86	38.65	165.49	0.15%	37.19%	13.65%	18.76%	48.89%	21.57%	
	40%	45.78	41.47	67.20	34.71	58.47	247.62	-0.12%	27.78%	10.13%	12.98%	37.50%	16.55%	
	50%	64.34	58.72	97.70	47.72	84.60	353.07	0.08%	19.60%	7.09%	8.61%	27.62%	12.14%	
	60%	88.17	81.26	140.23	63.63	120.91	494.19	0.31%	12.60%	4.28%	5.04%	18.59%	8.12%	
	70%	119.88	112.00	203.91	83.51	175.60	694.91	0.54%	6.29%	2.15%	2.40%	9.90%	4.41%	
	80%	164.36	156.58	310.70	109.07	270.63	1011.34	0.00%	0.25%	-0.11%	0.06%	-0.06%	0.09%	
	90%	230.37	226.84	530.92	143.16	500.05	1631.34	0.15%	-4.84%	-1.96%	-1.36%	-12.10%	5.41%	
	95%	277.48	280.23	766.86	164.85	872.88	2362.29	0.02%	-7.64%	-2.39%	-2.18%	-19.24%	9.70%	7.58%
Two-Point (70/80%) IR Method w/ Historical Data	10%	8.59	4.59	10.07	5.72	4.03	33.00	-0.15%	-4.19%	3.11%	12.40%	-31.02%	8.69%	
	20%	18.74	11.03	22.88	12.66	10.74	76.05	-0.03%	-2.77%	2.18%	8.50%	-24.29%	6.70%	
	30%	30.92	19.91	39.45	21.12	21.24	132.64	0.15%	-1.91%	1.38%	5.11%	-18.17%	4.93%	
	40%	45.78	32.07	61.42	31.51	37.23	208.01	-0.12%	-1.19%	0.65%	2.59%	-12.44%	3.26%	
	50%	64.34	48.81	91.46	44.43	61.51	310.55	0.08%	-0.58%	0.25%	1.12%	-7.21%	1.86%	
	60%	88.17	72.21	134.40	60.70	99.08	454.55	0.31%	0.06%	-0.06%	0.21%	-2.82%	0.74%	
	70%	119.24	105.56	199.68	81.59	159.95	666.02	0.00%	0.18%	0.03%	0.04%	0.10%	0.07%	
	80%	164.36	156.58	310.70	109.07	270.63	1011.34	0.00%	0.25%	-0.11%	0.06%	-0.06%	0.09%	
	90%	230.37	238.67	539.17	146.45	534.38	1689.04	0.15%	0.12%	-0.43%	0.91%	-6.06%	2.25%	
	95%	277.48	301.59	782.04	170.54	940.31	2471.96	0.02%	-0.59%	-0.46%	1.19%	-13.00%	5.66%	3.02%

The mean service times in Case B-7 are exactly the same as the mean service times in Case B-6. However, all service time SCVs are greater than 1. Since the ASIA system queueing time becomes a lower bound in this case, all intrinsic ratios are greater than 1. While the two-point IR method performs the best, all total absolute average errors are less than 15% in this case.

Case B-7 possesses the nearly-linear relationship of the intrinsic ratios as shown in Figure 6.10. However, for the fifth server, due to the sharp increasing of the intrinsic ratio in heavy traffic, the IR approaches have relatively larger errors in this situation. Nevertheless, the two-point IR method still gives the smallest errors among the five methods in heavy traffic.

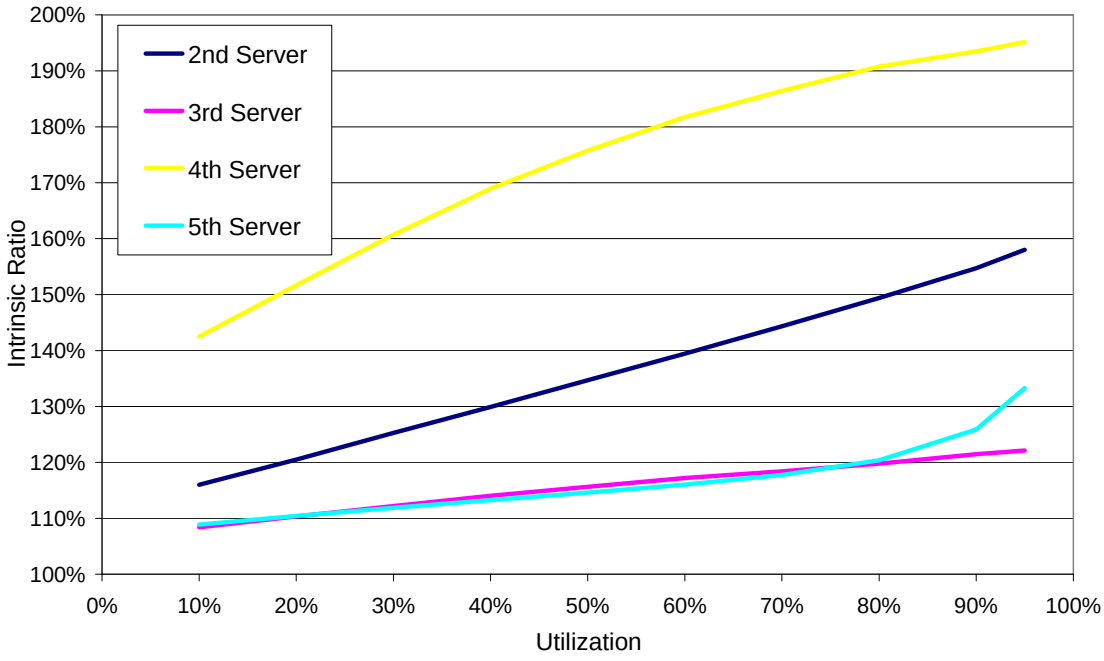

Figure 6.10 Intrinsic Ratios of Case B-7

Case B-8:

- Poisson arrivals

- Service time mean: (23, 25, 28, 20, 30) and SCV: (0.36, 0.25, 0.8, 0.5, 0.64)

Table 6.16 Queueing time approximations in Case B-8

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	1.30	1.04	2.04	0.87	2.33	7.58	0.15%	0.15%	0.18%	0.18%	0.16%		
	20%	2.83	2.27	4.55	1.86	5.23	16.74	0.12%	0.13%	0.16%	0.16%	0.15%		
	30%	4.67	3.77	7.70	3.02	8.95	28.11	0.11%	0.10%	0.15%	0.11%	0.12%		
	40%	6.93	5.64	11.81	4.38	13.95	42.71	0.12%	0.13%	0.15%	0.12%	0.15%		
	50%	9.71	8.01	17.35	6.03	21.01	62.11	0.12%	0.13%	0.20%	0.13%	0.18%		
	60%	13.33	11.20	25.34	8.03	31.62	89.52	0.13%	0.13%	0.20%	0.14%	0.18%		
	70%	18.12	15.71	37.84	10.57	49.70	131.93	0.14%	0.16%	0.21%	0.12%	0.23%		
	80%	24.84	22.52	60.05	13.84	86.30	207.54	0.14%	0.20%	0.31%	0.14%	0.28%		
	90%	34.74	34.13	110.43	18.33	202.18	399.82	0.17%	0.23%	0.42%	0.14%	0.58%		
	95%	41.88	43.60	168.14	21.25	439.16	714.03	0.19%	0.27%	0.57%	0.20%	1.22%		
QNA	10%	1.30	1.42	2.58	1.06	2.71	9.07	0.03%	36.42%	26.30%	22.66%	16.32%	19.70%	
	20%	2.83	3.09	5.67	2.24	5.96	19.80	0.07%	35.89%	24.69%	20.57%	13.98%	18.24%	
	30%	4.67	5.07	9.37	3.53	9.87	32.51	0.01%	34.58%	21.69%	17.02%	10.27%	15.67%	
	40%	6.92	7.44	13.87	4.92	14.69	47.84	-0.13%	31.93%	17.43%	12.39%	5.30%	12.06%	
	50%	9.72	10.32	19.50	6.47	21.02	67.03	0.09%	28.83%	12.38%	7.37%	0.05%	7.93%	
	60%	13.32	13.93	26.92	8.26	30.16	92.60	-0.02%	24.40%	6.22%	2.88%	-4.60%	6.70%	
	70%	18.12	18.65	37.55	10.49	45.34	130.14	-0.04%	18.73%	-0.76%	-0.78%	-8.77%	5.82%	
	80%	24.81	25.23	55.00	13.51	76.21	194.75	-0.11%	12.06%	-8.41%	-2.40%	-11.70%	8.78%	
	90%	34.81	35.45	91.49	17.93	170.79	350.47	0.20%	3.86%	-17.14%	-2.19%	-15.53%	13.04%	
	95%	41.93	43.25	131.79	20.99	361.62	599.58	0.11%	-0.80%	-21.62%	-1.21%	-17.66%	16.04%	12.80%
QNET	10%	1.30	0.97	1.93	0.90	2.14	7.23	0.03%	-7.03%	-5.77%	3.87%	-8.34%	5.53%	
	20%	2.83	2.13	4.30	1.94	4.81	16.01	0.07%	-6.39%	-5.46%	4.22%	-8.06%	5.35%	
	30%	4.67	3.55	7.29	3.15	8.25	26.92	0.01%	-5.68%	-5.29%	4.49%	-7.80%	5.18%	
	40%	6.92	5.34	11.19	4.58	12.85	40.88	-0.13%	-5.29%	-5.27%	4.59%	-7.93%	5.24%	
	50%	9.72	7.65	16.46	6.30	19.28	59.41	0.09%	-4.55%	-5.16%	4.56%	-8.22%	5.27%	
	60%	13.32	10.74	23.96	8.40	28.94	85.36	-0.02%	-4.13%	-5.48%	4.61%	-8.47%	5.48%	
	70%	18.12	15.08	35.44	11.02	45.09	124.75	-0.04%	-3.96%	-6.34%	4.27%	-9.27%	6.13%	
	80%	24.81	21.64	55.12	14.41	77.74	193.73	-0.11%	-3.88%	-8.20%	4.14%	-9.91%	7.21%	
	90%	34.81	32.64	96.57	19.02	179.44	362.48	0.20%	-4.37%	-12.55%	3.75%	-11.25%	9.72%	
	95%	41.93	41.47	141.86	22.07	389.94	637.26	0.11%	-4.88%	-15.63%	3.87%	-11.21%	10.99%	9.03%
IR Method with QNA	10%	1.30	1.11	1.67	0.89	1.80	6.76	0.03%	6.48%	-18.37%	2.39%	-22.60%	13.07%	
	20%	2.83	2.44	3.75	1.91	4.10	15.04	0.07%	7.30%	-17.45%	2.75%	-21.61%	12.80%	
	30%	4.67	4.07	6.43	3.11	7.12	25.41	0.01%	8.23%	-16.50%	3.03%	-20.46%	12.47%	
	40%	6.92	6.13	9.99	4.52	11.24	38.80	-0.13%	8.83%	-15.47%	3.17%	-19.43%	12.14%	
	50%	9.72	8.80	14.92	6.22	17.18	56.83	0.09%	9.87%	-14.05%	3.17%	-18.24%	11.69%	
	60%	13.32	12.39	22.17	8.29	26.38	82.56	-0.02%	10.65%	-12.53%	3.24%	-16.56%	11.02%	
	70%	18.12	17.48	33.78	10.88	42.37	122.63	-0.04%	11.29%	-10.72%	2.91%	-14.73%	10.20%	
	80%	24.81	25.23	55.00	14.21	76.07	195.32	-0.11%	12.06%	-8.41%	2.67%	-11.85%	8.86%	
	90%	34.81	38.43	104.09	18.65	184.71	380.69	0.20%	12.59%	-5.74%	1.71%	-8.64%	7.13%	
	95%	41.93	49.20	162.05	21.47	415.98	690.64	0.11%	12.86%	-3.62%	1.06%	-5.28%	4.92%	7.32%
Single-Point (80%) IR Method w/ Historical Data	10%	1.30	0.97	1.92	0.87	2.25	7.30	0.03%	-6.97%	-6.14%	-0.13%	-3.48%	3.70%	
	20%	2.83	2.13	4.30	1.87	5.09	16.22	0.07%	-6.10%	-5.40%	0.22%	-2.83%	3.22%	
	30%	4.67	3.57	7.34	3.03	8.76	27.38	0.01%	-5.12%	-4.70%	0.50%	-2.11%	2.70%	
	40%	6.92	5.39	11.34	4.41	13.72	41.77	-0.13%	-4.37%	-4.01%	0.63%	-1.69%	2.32%	
	50%	9.72	7.76	16.83	6.06	20.74	61.11	0.09%	-3.18%	-3.00%	0.63%	-1.28%	1.76%	
	60%	13.32	10.96	24.83	8.08	31.43	88.62	-0.02%	-2.14%	-2.05%	0.71%	-0.60%	1.13%	
	70%	18.12	15.53	37.45	10.61	49.58	131.29	-0.04%	-1.12%	-1.02%	0.38%	-0.22%	0.55%	
	80%	24.84	22.55	60.13	13.86	86.79	208.17	0.00%	0.17%	0.14%	0.15%	0.57%	0.30%	
	90%	34.81	34.69	111.56	18.19	202.34	401.59	0.20%	1.62%	1.02%	-0.79%	0.08%	0.51%	
	95%	41.93	44.69	171.30	20.95	440.73	719.60	0.11%	2.52%	1.88%	-1.42%	0.36%	0.86%	0.84%
Two-Point (70/80%) IR Method w/ Historical Data	10%	1.30	1.05	2.13	0.83	2.26	7.57	0.03%	1.40%	4.00%	-3.90%	-2.98%	2.64%	
	20%	2.83	2.30	4.69	1.81	5.11	16.73	0.07%	1.05%	3.11%	-3.02%	-2.41%	2.09%	
	30%	4.67	3.80	7.87	2.95	8.79	28.08	0.01%	0.82%	2.21%	-2.20%	-1.77%	1.52%	
	40%	6.92	5.65	11.97	4.31	13.75	42.61	-0.13%	0.33%	1.35%	-1.54%	-1.42%	1.06%	
	50%	9.72	8.04	17.50	5.97	20.78	62.01	0.09%	0.30%	0.87%	-0.99%	-1.09%	0.76%	
	60%	13.32	11.21	25.44	8.00	31.47	89.45	-0.02%	0.13%	0.39%	-0.38%	-0.48%	0.33%	
	70%	18.12	15.70	37.88	10.55	49.61	131.87	0.00%	-0.03%	0.10%	-0.16%	-0.17%	0.11%	
	80%	24.84	22.55	60.13	13.86	86.79	208.17	0.00%	0.17%	0.14%	0.15%	0.57%	0.30%	
	90%	34.81	34.35	110.68	18.29	202.25	400.38	0.20%	0.65%	0.23%	-0.25%	0.04%	0.16%	
	95%	41.93	44.09	169.65	21.12	440.51	717.30	0.11%	1.14%	0.90%	-0.62%	0.31%	0.50%	0.42%

In Case B-8, the mean service times are the same as the previous two cases, but all service time SCVs are smaller than 1. The total average absolute errors of all five methods are than 13%. Among the five, the two IR methods with historical data perform the best, and give less than 0.5% errors. Compared with IR based approaches, both QNA and QNET give larger errors in heavy traffic.

Case B-8 possesses the nearly-linear relationship of the intrinsic ratios as shown in Figure 6.11.

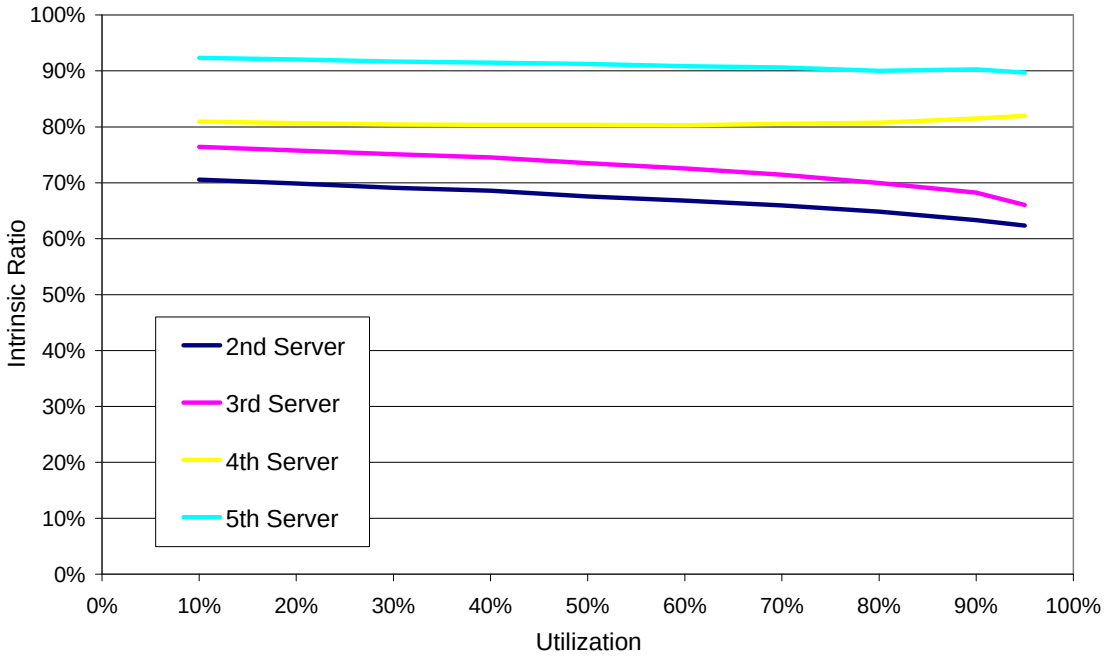

Figure 6.11 Intrinsic Ratios of Case B-8

Case B-9:

- Arrival process: Gamma distribution with SCV = 0.1 (i.e. Erlang(10))

- Service time mean: (23, 25, 28, 20, 30) and SCV: (0.36, 0.25, 0.8, 0.5, 0.64)

Table 6.17 Queueing time approximations in Case B-9

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	0.00	0.00	0.02	0.01	0.05	0.08	17.88%	9.14%	1.90%	2.16%	1.02%		
	20%	0.01	0.03	0.42	0.22	0.93	1.61	1.91%	0.97%	0.42%	0.37%	0.26%		
	30%	0.07	0.22	1.71	0.85	3.11	5.97	0.63%	0.38%	0.26%	0.23%	0.14%		
	40%	0.27	0.72	4.01	1.87	6.57	13.45	0.37%	0.24%	0.18%	0.16%	0.16%		
	50%	0.70	1.62	7.48	3.26	11.61	24.66	0.26%	0.16%	0.18%	0.15%	0.16%		
	60%	1.44	3.02	12.64	5.08	19.33	41.51	0.21%	0.14%	0.19%	0.15%	0.16%		
	70%	2.63	5.12	20.65	7.48	31.97	67.85	0.20%	0.13%	0.18%	0.13%	0.17%		
	80%	4.51	8.39	34.71	10.77	56.99	115.36	0.16%	0.14%	0.23%	0.13%	0.19%		
	90%	7.53	13.89	64.63	15.46	129.00	230.52	0.16%	0.13%	0.28%	0.14%	0.34%		
	95%	9.79	18.28	98.30	18.73	264.30	409.40	0.18%	0.17%	0.38%	0.13%	0.66%		
QNA	10%	0.44	0.40	1.30	0.43	1.25	3.82	611744%	115235%	8146.14%	4556.05%	2366.07%	4916.65%	
	20%	0.96	0.89	2.92	0.98	2.93	8.68	12203.9%	2838.3%	601.23%	337.28%	215.65%	440.32%	
	30%	1.58	1.52	5.02	1.69	5.33	15.13	2128.06%	579.57%	192.82%	98.29%	71.09%	153.33%	
	40%	2.34	2.34	7.83	2.66	8.90	24.06	756.73%	223.40%	95.04%	42.04%	35.50%	78.94%	
	50%	3.29	3.47	11.73	3.99	14.40	36.87	371.67%	114.31%	56.81%	22.43%	24.01%	49.50%	
	60%	4.51	5.06	17.44	5.82	23.26	56.09	212.60%	67.73%	38.01%	14.66%	20.33%	35.14%	
	70%	6.13	7.44	26.40	8.37	38.71	87.03	132.87%	45.14%	27.84%	11.89%	21.08%	28.27%	
	80%	8.39	11.20	42.13	11.93	70.28	143.93	86.19%	33.51%	21.39%	10.77%	23.32%	24.76%	
	90%	11.77	17.77	76.33	17.02	165.54	288.43	56.29%	27.93%	18.10%	10.07%	28.32%	25.12%	
	95%	14.18	23.18	114.51	20.41	355.87	528.15	44.82%	26.80%	16.49%	8.94%	34.65%	29.01%	30.95%
QNET	10%	0.44	0.55	1.42	0.77	1.82	5.01	611686%	159131%	8930.04%	8155.27%	3489.01%	6465.90%	
	20%	0.96	1.21	3.17	1.67	4.10	11.11	12203.5%	3899.75%	660.66%	647.48%	341.76%	591.73%	
	30%	1.58	2.02	5.37	2.72	7.03	18.72	2128.02%	804.37%	213.02%	219.95%	125.73%	213.34%	
	40%	2.34	3.02	8.21	3.99	10.93	28.49	756.73%	317.36%	104.56%	113.27%	66.47%	111.87%	
	50%	3.29	4.31	12.03	5.53	16.39	41.55	371.67%	166.21%	60.80%	69.78%	41.18%	68.45%	
	60%	4.51	6.01	17.45	7.45	24.56	59.97	212.60%	99.16%	38.05%	46.67%	27.04%	44.49%	
	70%	6.13	8.38	25.73	9.91	38.06	88.22	132.88%	63.67%	24.59%	32.56%	19.07%	30.02%	
	80%	8.39	11.92	40.00	13.20	64.61	138.12	86.19%	42.10%	15.24%	22.59%	13.37%	19.73%	
	90%	11.77	17.73	70.67	17.85	140.53	258.56	56.29%	27.68%	9.34%	15.41%	8.94%	12.16%	
	95%	14.18	22.33	104.83	21.01	281.64	443.98	44.82%	22.19%	6.64%	12.14%	6.56%	8.45%	20.13%
IR Method with QNA	10%	0.44	0.53	1.54	0.88	2.42	5.81	611744%	151689%	9682.23%	9296.14%	4677.89%	7518.28%	
	20%	0.96	1.15	3.43	1.89	5.39	12.82	12204%	3709.22%	722.56%	746.95%	480.83%	698.36%	
	30%	1.58	1.92	5.79	3.07	9.10	21.46	2128.06%	760.33%	237.77%	260.59%	192.40%	259.34%	
	40%	2.34	2.87	8.83	4.47	13.90	32.41	756.73%	296.50%	120.15%	138.79%	111.75%	141.06%	
	50%	3.29	4.08	12.90	6.14	20.38	46.80	371.67%	152.46%	72.50%	88.56%	75.54%	89.75%	
	60%	4.51	5.69	18.64	8.19	29.66	66.68	212.60%	88.45%	47.47%	61.21%	53.42%	60.64%	
	70%	6.13	7.91	27.32	10.75	44.21	96.32	132.87%	54.41%	32.33%	43.70%	38.30%	41.96%	
	80%	8.39	11.20	42.13	14.04	71.02	146.77	86.19%	33.51%	21.39%	30.34%	24.62%	27.23%	
	90%	11.77	16.56	73.37	18.42	142.12	262.24	56.29%	19.21%	13.52%	19.14%	10.17%	13.76%	
	95%	14.18	20.76	107.48	21.22	268.41	432.05	44.82%	13.57%	9.34%	13.26%	1.55%	5.53%	23.39%
Single-Point (80%) IR Method w/ Historical Data	10%	0.44	0.36	1.14	0.67	1.79	4.41	611744%	104659%	7155.52%	7108.24%	3434.05%	5686.59%	
	20%	0.96	0.80	2.56	1.45	4.00	9.77	12203.9%	2536.68%	513.30%	549.74%	331.81%	508.41%	
	30%	1.58	1.33	4.35	2.36	6.81	16.42	2128.06%	497.60%	153.44%	176.63%	118.74%	174.96%	
	40%	2.34	2.00	6.68	3.43	10.48	24.93	756.73%	176.60%	66.48%	83.19%	59.63%	85.40%	
	50%	3.29	2.86	9.85	4.71	15.52	36.24	371.67%	77.07%	31.74%	44.65%	33.66%	46.93%	
	60%	4.51	4.02	14.42	6.28	22.88	52.10	212.60%	33.09%	14.07%	23.67%	18.38%	25.53%	
	70%	6.13	5.64	21.51	8.24	34.76	76.28	132.87%	10.06%	4.16%	10.24%	8.73%	12.42%	
	80%	4.51	8.39	34.68	10.77	56.85	115.20	0.00%	0.08%	-0.07%	-0.01%	-0.25%	0.15%	
	90%	11.77	12.19	61.59	14.13	121.17	220.86	56.29%	-12.23%	-4.71%	-8.60%	-6.07%	7.87%	
	95%	14.18	15.50	92.93	16.28	240.53	379.42	44.82%	-15.20%	-5.46%	-13.11%	-8.99%	9.47%	13.41%
Two-Point (70/80%) IR Method w/ Historical Data	10%	0.44	-0.57	1.51	0.24	1.09	2.70	611744%	-163889%	9450.17%	2442.76%	2045.75%	4935.61%	
	20%	0.96	-0.94	2.87	0.65	2.61	6.14	12203.9%	-3202.3%	587.26%	189.28%	181.51%	403.27%	
	30%	1.58	-1.06	4.26	1.27	4.74	10.79	2128.06%	-574.02%	148.48%	48.74%	52.21%	123.49%	
	40%	2.34	-0.83	5.97	2.16	7.79	17.43	756.73%	-214.64%	48.82%	15.44%	18.58%	52.71%	
	50%	3.29	-0.12	8.46	3.40	12.35	27.39	371.67%	-107.45%	13.10%	4.53%	6.39%	25.13%	
	60%	4.51	1.29	12.54	5.12	19.62	43.08	212.60%	-57.24%	-0.76%	0.80%	1.51%	12.58%	
	70%	2.63	5.12	20.67	7.48	32.01	67.91	0.00%	-0.06%	0.11%	0.04%	0.12%	0.10%	
	80%	4.51	8.39	34.68	10.77	56.85	115.20	0.00%	0.08%	-0.07%	-0.01%	-0.25%	0.15%	
	90%	11.77	15.75	66.60	15.44	128.02	237.58	56.29%	13.41%	3.03%	-0.15%	-0.76%	3.93%	
	95%	14.18	21.93	102.89	18.53	254.75	412.28	44.82%	20.01%	4.67%	-1.06%	-3.61%	5.47%	7.45%

The conditions in Case B-9 are almost the same as the conditions in Case B-8, except for the arrival process. In Case B-9, the arrival process follows a gamma distribution and the SCV is 0.1. Since the arrival process is not Poisson, Kingman's approximation is not exact anymore. In contrast to the previous cases, the ASIA system queueing times cannot be calculated exactly.

From Figure 5.1 (in Section 5.2), we know Kingman's approximation over-estimates the true values. The upper and lower bounds are over-estimated, especially in light traffic. When a server is the bottleneck in a sub-system, its lower bound is the (inexact) ASIA system queueing time minus the simulation queueing times of the previous servers. Figure 6.12 shows the intrinsic ratios are below 0 (i.e. the over-estimated lower bound) for the second, third and fifth servers in light traffic except for the forth server, since its lower bound is 0, which is exact. Although all five methods perform poorly in this case, two-point IR method still gives the smallest errors.

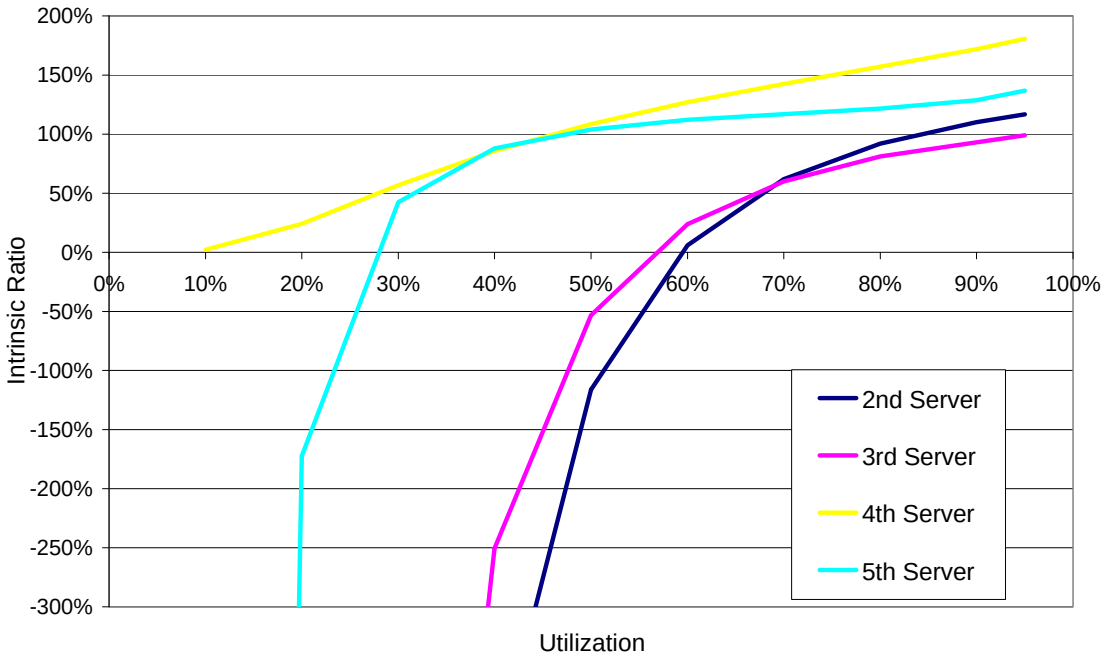

Figure 6.12 Intrinsic Ratios of Case B-9

Case B-10:

- Arrival process: Gamma distribution with SCV = 5

- Service time mean: (23, 25, 28, 20, 30) and SCV: (0.36, 0.25, 0.8, 0.5, 0.64)

Table 6.18 Queueing time approximations in Case B-10

	BN Util.	QT 1	QT 2	QT 3	QT 4	QT 5	Sys. QT	Error 1	Error 2	Error 3	Error 4	Error 5	Avg. %	TTL Avg
Simulation	10%	20.83	8.86	12.63	4.39	12.03	58.74	0.09%	0.11%	0.12%	0.12%	0.11%		
	20%	30.04	12.86	18.99	6.13	18.51	86.53	0.09%	0.10%	0.12%	0.12%	0.13%		
	30%	39.35	17.06	26.10	7.80	26.01	116.32	0.09%	0.13%	0.18%	0.13%	0.16%		
	40%	49.66	21.96	34.94	9.53	35.70	151.80	0.11%	0.14%	0.20%	0.13%	0.18%		
	50%	61.95	28.18	47.05	11.44	49.48	198.10	0.12%	0.16%	0.21%	0.13%	0.22%		
	60%	77.33	36.42	64.62	13.62	70.93	262.92	0.15%	0.20%	0.28%	0.14%	0.29%		
	70%	97.03	48.05	92.89	16.14	108.02	362.12	0.17%	0.24%	0.31%	0.16%	0.37%		
	80%	124.08	66.34	145.84	19.18	190.50	545.95	0.25%	0.31%	0.46%	0.18%	0.62%		
	90%	164.41	100.41	278.42	22.85	480.93	1047.02	0.26%	0.46%	0.74%	0.16%	1.26%		
	95%	193.72	130.14	442.15	25.07	1174.26	1965.34	0.37%	0.55%	0.97%	0.18%	2.13%		
QNA	10%	5.12	5.93	8.27	3.86	9.21	32.39	-75.43%	-32.99%	-34.53%	-12.07%	-23.46%	44.86%	
	20%	11.16	12.85	17.87	7.88	19.46	69.23	-62.83%	-0.08%	-5.90%	28.64%	5.16%	26.25%	
	30%	18.41	20.85	28.71	11.71	30.07	109.76	-53.21%	22.22%	9.98%	50.24%	15.60%	30.35%	
	40%	27.26	30.09	40.74	15.01	40.44	153.53	-45.10%	37.02%	16.58%	57.47%	13.27%	30.66%	
	50%	38.32	40.79	54.05	17.50	50.43	201.07	-38.15%	44.76%	14.87%	52.97%	1.91%	25.36%	
	60%	52.51	53.35	69.07	19.08	60.85	254.86	-32.10%	46.49%	6.87%	40.12%	-14.21%	23.48%	
	70%	71.40	68.49	87.12	19.91	74.82	321.74	-26.42%	42.54%	-6.20%	23.34%	-30.73%	24.52%	
	80%	97.77	87.61	112.20	20.52	102.54	420.65	-21.20%	32.07%	-23.07%	6.97%	-46.17%	31.23%	
	90%	137.20	114.03	158.88	21.98	194.11	626.20	-16.55%	13.57%	-42.93%	-3.80%	-59.64%	42.79%	
	95%	165.26	132.46	208.57	23.58	387.21	917.08	-14.69%	1.78%	-52.83%	-5.94%	-67.03%	53.57%	41.78%
QNET	10%	5.12	1.95	2.25	0.85	2.21	12.39	-75.43%	-77.95%	-82.18%	-80.65%	-81.59%	78.91%	
	20%	11.16	4.37	4.94	1.82	4.92	27.21	-62.84%	-66.01%	-74.00%	-70.38%	-73.43%	68.56%	
	30%	18.41	7.42	8.21	2.93	8.30	45.27	-53.21%	-56.50%	-68.55%	-62.47%	-68.09%	61.08%	
	40%	27.26	11.33	12.30	4.22	12.69	67.79	-45.10%	-48.42%	-64.81%	-55.70%	-64.47%	55.34%	
	50%	38.32	16.41	17.56	5.76	18.66	96.71	-38.15%	-41.76%	-62.67%	-49.63%	-62.30%	51.18%	
	60%	52.51	23.12	24.59	7.65	27.38	135.24	-32.10%	-36.52%	-61.95%	-43.86%	-61.40%	48.56%	
	70%	71.40	32.12	34.42	10.04	41.55	189.53	-26.42%	-33.15%	-62.94%	-37.79%	-61.53%	47.66%	
	80%	97.77	44.22	49.14	13.28	69.07	273.48	-21.20%	-33.34%	-66.31%	-30.77%	-63.74%	49.91%	
	90%	137.20	59.85	72.91	18.03	147.34	435.33	-16.55%	-40.40%	-73.81%	-21.09%	-69.36%	58.42%	
	95%	165.25	68.59	91.64	21.44	257.70	604.62	-14.69%	-47.30%	-79.27%	-14.46%	-78.05%	69.24%	60.63%
IR Method with QNA	10%	5.12	3.68	2.32	0.94	0.46	12.52	-75.43%	-58.43%	-81.60%	-78.69%	-96.19%	78.69%	
	20%	11.16	8.14	5.40	2.01	1.34	28.06	-62.83%	-36.70%	-71.58%	-67.13%	-92.76%	67.57%	
	30%	18.41	13.66	9.59	3.27	2.94	47.87	-53.21%	-19.95%	-63.27%	-58.01%	-88.68%	58.84%	
	40%	27.26	20.64	15.52	4.76	5.84	74.03	-45.10%	-5.98%	-55.58%	-50.03%	-83.65%	51.23%	
	50%	38.32	29.77	24.36	6.55	11.20	110.20	-38.15%	5.67%	-48.23%	-42.76%	-77.37%	45.99%	
	60%	52.51	42.19	38.41	8.73	21.74	163.57	-32.10%	15.84%	-40.57%	-35.90%	-69.36%	42.18%	
	70%	71.40	60.01	62.92	11.46	44.82	250.61	-26.42%	24.90%	-32.26%	-29.02%	-58.51%	37.40%	
	80%	97.77	87.61	112.20	14.97	106.66	419.21	-21.20%	32.07%	-23.07%	-21.97%	-44.01%	31.01%	
	90%	137.20	135.64	239.21	19.64	367.18	898.87	-16.55%	35.09%	-14.08%	-14.03%	-23.65%	20.88%	
	95%	165.26	175.62	401.52	22.62	1040.18	1805.20	-14.69%	34.95%	-9.19%	-9.77%	-11.42%	12.78%	25.60%
Single-Point (80%) IR Method w/ Historical Data	10%	5.12	3.30	4.24	1.20	3.66	17.53	-75.43%	-62.70%	-66.43%	-72.70%	-69.55%	70.16%	
	20%	11.16	7.32	9.60	2.58	8.46	39.13	-62.83%	-43.10%	-49.44%	-57.87%	-54.29%	54.78%	
	30%	18.41	12.30	16.57	4.20	14.96	66.43	-53.21%	-27.92%	-36.54%	-46.19%	-42.50%	42.89%	
	40%	27.26	18.63	25.94	6.10	24.15	102.08	-45.10%	-15.15%	-25.76%	-35.97%	-32.37%	32.75%	
	50%	38.32	26.94	39.15	8.39	37.92	150.72	-38.15%	-4.37%	-16.79%	-26.65%	-23.37%	23.92%	
	60%	52.51	38.31	58.95	11.19	60.29	221.25	-32.10%	5.19%	-8.78%	-17.86%	-15.00%	17.29%	
	70%	71.40	54.74	91.37	14.68	101.31	333.50	-26.42%	13.93%	-1.63%	-9.05%	-6.21%	11.60%	
	80%	124.08	66.71	146.06	19.18	191.06	547.10	0.00%	0.55%	0.15%	-0.01%	0.29%	0.21%	
	90%	137.20	125.51	297.88	25.17	518.99	1104.76	-16.55%	25.00%	6.99%	10.16%	7.91%	10.71%	
	95%	165.26	163.42	474.61	28.98	1262.98	2095.25	-14.69%	25.57%	7.34%	15.63%	7.56%	9.51%	13.00%
Two-Point (70/80%) IR Method w/ Historical Data	10%	5.12	5.78	8.73	2.04	10.29	31.97	-75.43%	-34.68%	-30.90%	-53.44%	-14.43%	45.57%	
	20%	11.16	11.96	17.70	4.14	20.08	65.04	-62.83%	-7.04%	-6.82%	-32.41%	8.49%	28.47%	
	30%	18.41	18.67	27.30	6.31	29.98	100.68	-53.21%	9.45%	4.59%	-19.08%	15.26%	25.11%	
	40%	27.26	26.18	38.22	8.56	40.99	141.21	-45.10%	19.25%	9.38%	-10.16%	14.80%	23.82%	
	50%	38.32	34.91	51.67	10.93	54.88	190.70	-38.15%	23.88%	9.82%	-4.48%	10.91%	20.65%	
	60%	52.51	45.59	70.05	13.44	75.36	256.95	-32.10%	25.16%	8.40%	-1.31%	6.25%	16.75%	
	70%	97.03	48.13	92.92	16.16	108.72	362.95	0.00%	0.16%	0.04%	0.12%	0.65%	0.23%	
	80%	124.08	66.71	146.06	19.18	191.06	547.10	0.00%	0.55%	0.15%	-0.01%	0.29%	0.21%	
	90%	137.20	116.01	283.61	22.63	493.51	1052.97	-16.55%	15.54%	1.87%	-0.93%	2.62%	5.81%	
	95%	165.26	146.25	448.10	24.60	1206.40	1990.61	-14.69%	12.38%	1.35%	-1.84%	2.74%	4.23%	7.25%

The conditions in Case B-10 are almost the same as the conditions in Case B-8 and 9, except that the arrival process SCV is 5. In this case, Kingman's heavy traffic approximation tends to under-estimate the true values. We get an under-estimated upper and lower bounds in light traffic. When a server is a bottleneck in a sub-system, the lower bound is its (inexact) ASIA system queueing time minus the simulation queueing times of previous servers. The intrinsic ratios are above one (i.e. the under-estimated upper bound) in light traffic for all four servers as shown in Figure 6.13.

Case B-10 still possesses the nearly-linear relationship of the intrinsic ratios as shown in Figure 6.13. All five methods perform poorly in this case, while QNET gives error more than 60%. The two-point IR method still performs the best.

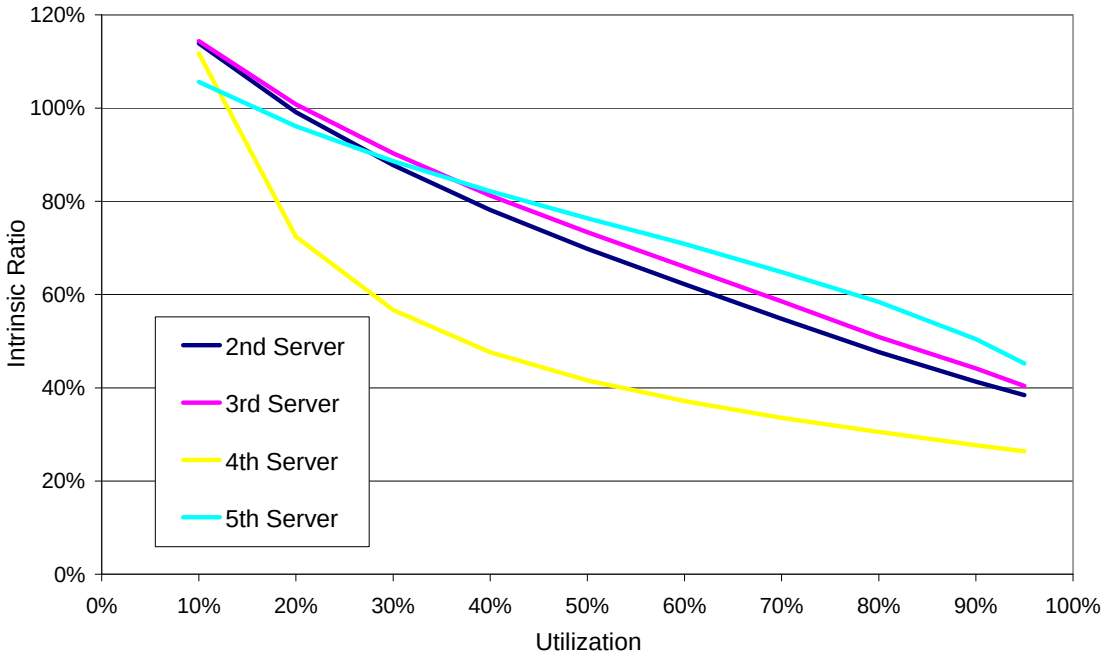

Figure 6.13 Intrinsic Ratios of Case B-10

We now summarize the insights obtained from the extensive set of experiments presented above. In all 10 examined cases, the two-point IR method with historical data always performs the best, and the one-point IR method with historical data is the second in nine cases. From the smallest to the largest, the total average errors from all 10 cases are 2.7%, 5.6%, 11.3%, 19.6% and 19.7% for the two-point IR method, one-point IR method, IR method with QNA, QNA, and QNET, respectively. Considering the confidence intervals of simulations, the 2.7% error from two-point IR method is indeed an impressive result. It should be noted all intrinsic ratio approaches perform well in heavy traffic, but QNA and QNET do not. QNA performs the best at around 60 to 80% utilization. The average errors from QNA and QNET are about the same. However, in terms of the range of average errors, QNA ranges from 8.0 to 41.8%, while QNET ranges from 3.3 to 60.6%. The performance of QNET is more unstable than QNA.

In the examined cases, when service time SCV is small and bottlenecks are not the first server, 20% to 30% errors are observed from QNA and QNET. Furthermore, the percentage errors of QNA, QNET and IR method with QNA tend to increase at the downstream servers, which means the total average errors could be even larger when the sequence is longer. However, this tendency is less obvious in the IR methods with historical data, since their errors mainly come from the deviations of the nearly-linear relationship of the intrinsic ratios, which is calculated based on the historical data.

The IR methods with historical data perform very well when the utilizations are close to the historical utilizations, since it has a better prediction about the intrinsic ratio. In practice, we also care more about the performance in this region, since utilization usually changes gradually rather than dramatically. Because the errors from single-point and two-point IR methods with historical data are smaller and both are easier to implement, they give us better alternatives to QNA and QNET.

6.6 Dependence among General Queues in Series

The correlation among workstations is one key reason that predicting the behavior of a general Jackson queueing network is difficult. In this section, we explore the correlation from the viewpoint of ASIA systems. Using Procedure 6.2 and Eq. (6.26), we can describe the correlation quantitatively.

Procedure 6.2 tells us the contribution factor (f_i) of the system bottleneck term in Eq. (6.26) is always 1, which means the contribution from the system bottleneck to the system queueing time is non-discounted. The contribution is always the same as its queueing time in the corresponding ASIA system.

$$\sum_{i=1}^{n} QT_i = f_1\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + f_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \ldots + f_n\alpha_n\left(\frac{\rho_n}{1-\rho_n}\right)\frac{1}{\mu_n}, \qquad (6.26)$$

In addition to the bottleneck, to analyze the correlations among servers, it is necessary to know the contribution factor (f_i) of each server. Based on the simulated queueing time and Procedure 6.1, we can solve for x_i (or y_i). Based on the solved values of x_i (or y_i) and Procedure 6.2, we can solve for f_i of each server.

When the dispatching rule does not take the downstream servers into account, the behavior of the downstream servers has no impact on the upstream servers. In this situation, we have the following property:

Property 6.1 (Uni-Direction Property):
If the dispatching rule does not take the status of the downstream servers into account, queueing times of the upstream servers are independent of their downstream servers.

Proof: When the number of servers and buffer sizes are fixed, queueing time is determined by the arrival process and service time distribution. Since both of them are independent of the downstream servers when dispatching rules are independent of the

downstream servers, the queueing time of a server is independent of its downstream servers.

In other words, because the correlation among servers is caused by the non-renewal departure processes, and the departure processes are independent of the status of the downstream servers, the correlation among servers is uni-direction instead of bi-directions. The queueing time of an upstream server should not be affected by how many downstream servers it has or their status, if the dispatching rule does not take the downstream servers into account. This obvious property is possessed by QNA, but not QNET. The queueing times approximated by QNET may have different values when the status of downstream servers changes.

Based on Property 6.1, we can analyze the system behavior of tandem queues from the first server to the last server by gradually adding more servers. For example, if the system is only composed of one single server, we can analyze its performance by the models introduced in Part I. If another server is added to the system, the second queueing time is purely determined by the first and second servers. The actual queueing time of the first server should not change due to the extra downstream server. However, the value of f_1 would change if the second server has higher utilization, which means its contribution to the system queueing time changes.

Similarly, if we add the third server to the system, the first and second queueing times should not change, and the third queueing time is purely determined by the first, second and third servers. The contribution from the first and second servers (i.e. f_1 and f_2) to the system queueing time would change again, but it is purely the results of adding the third server. The analysis can be extended to more servers in the sequence.

Based on Property 6.1 and Definition 6.1, we can analyze queueing time dependence beginning by assuming n is 1 in Eq. (6.26), and then gradually increase n. Furthermore, instead of adding servers one by one, we can add one sub-system at a time,

where each sub-system contains all the currently considered servers to the next bottleneck, and each bottleneck is identified by the stage I of Procedure 6.2.

As we will see in the following, the effects from the upstream servers can be blocking or diffusion. Two examples are given in Section 6.6.1 and 6.6.2.

6.6.1 Blocking Effect

To see the blocking effects, we begin by studying the case where the service time SCV is smaller than the initial inter-arrival time SCV. The example of Case B-2 is used when its arrival rate is 37.5 (i.e. 80% system utilization). Since the initial arrival process is Poisson and all service time SCVs are 0.25, the internal inter-arrival time SCV between any two consecutive servers is always smaller than the initial arrival process SCV.

We begin by looking at the first sub-system of Case B-2, which is composed of the first two servers. Their service times are 25 and 28 and service time SCVs are 0.25. Based on the simulation results in Table 6.10, the queueing times are 31.27 and 36.43 (the numbers with underlines in Figure 6.14). Since y_2 is 0.516, f_1 is 0.516 and f_2 is 1. Based on the P-K formula, the ASIA system queueing time of the second server is 51.58. Therefore, the contributions from the first and second servers to the system queueing time are 16.12 (=31.27* f_1) and 51.58 (=51.58* f_2), respectively. It should be noted that we use the simulated queueing time (31.27) instead of the ASIA system queueing time (31.25) for the first server.

If the second server sees the initial arrival process, the queueing time (i.e. the bottleneck in this sub-system) should be 51.58. However, it is only 36.43. The difference (of 15.15) between its true queueing time (i.e. 36.43) and its ASIA system queueing time (i.e. 51.58) is the result of blocking at the first server as shown in Figure 6.14. In general, the ASIA system queueing time can be decomposed into two portions: The distributed part and the non-distributed part. The distributed part explains the correlations among

servers and is the result of the fully coupled system. The non-distributed part is the portion of ASIA system queueing time which is not distributed to other servers. When all service times are constant, based on the reduction method, the observed queueing time at all the non-bottleneck stations consists of only what is distributed from the bottleneck station. In general, the non-distributed portion of the ASIA system queueing time is called the base queueing time as shown in Figure 6.14.

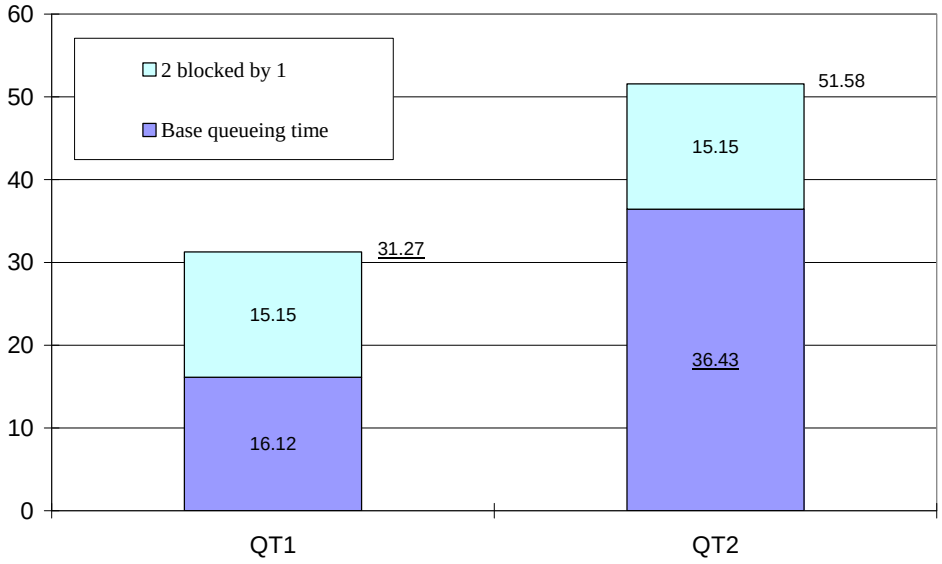

Figure 6.14 Queueing time analysis of the first sub-system in Case B-2

Now consider the second sub-system composed of the first three servers in Case B-2. The service times of the first three servers are 25, 28 and 30 and the SCVs are 0.25 for all servers. Based on Table 6.10, the queueing times are 31.27, 36.43 and 48.29 (the numbers with underlines in Figure 6.15). Since y_2 is 0.516 and y_3 is 0.605, f_1 is 0.312, f_2 is 0.605 and f_3 is 1. The ASIA system queueing times of the second and the third server are 51.58 and 75, respectively. The contributions from the first, second and the third server to the system queueing time are 9.75, 31.23 and 75, respectively.

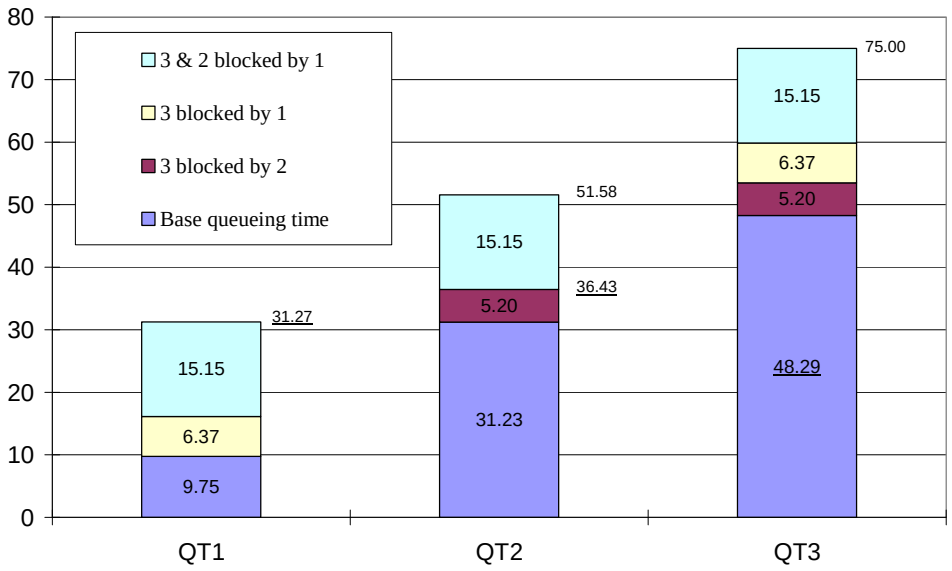

Figure 6.15 Queueing time analysis of the second sub-system in Case B-2

In the second sub-system, the contribution to system queueing time from the first server has been dropped to from 16.12 to 9.75. The difference (of 6.37) comes from blocking the queueing time of the third server. Similarly, the contribution from the second server is only 31.23. In addition to the portion of 15.15, which has been blocked by the first server, the difference (of 5.20) comes from blocking the queueing time of the third server. Therefore, although the contribution of the bottleneck to the system queueing time is 75 (because f_3 is 1 in Eq. (6.26)), we only see 48.29 (the actual queueing time). All the other parts have been blocked by the first and second servers.

The third system considers the original system which contains all 5 servers in Case B-2. The service times of the five servers are 25, 28, 30, 20 and 25 and the SCVs are 0.25 for all servers. Since the forth and fifth servers are non-bottlenecks, the coefficients x_4 and x_5 have no impact on the previous 3 servers. Based on Table 6.10, the forth and fifth queueing times are 4.35 and 13.32. Since x_4 is 0.304 and x_5 is 0.426, f_4 is 0.304 and f_5 is 0.426. Their ASIA system queueing times are 14.29 and 31.25. The contributions from the forth and fifth servers to the system queueing time are 4.35 and

13.32. It should be noted, unlike bottlenecks, the blocked portion of the non-bottleneck server does not decrease the contribution of the previous bottleneck. The contribution of the third server is still 75.

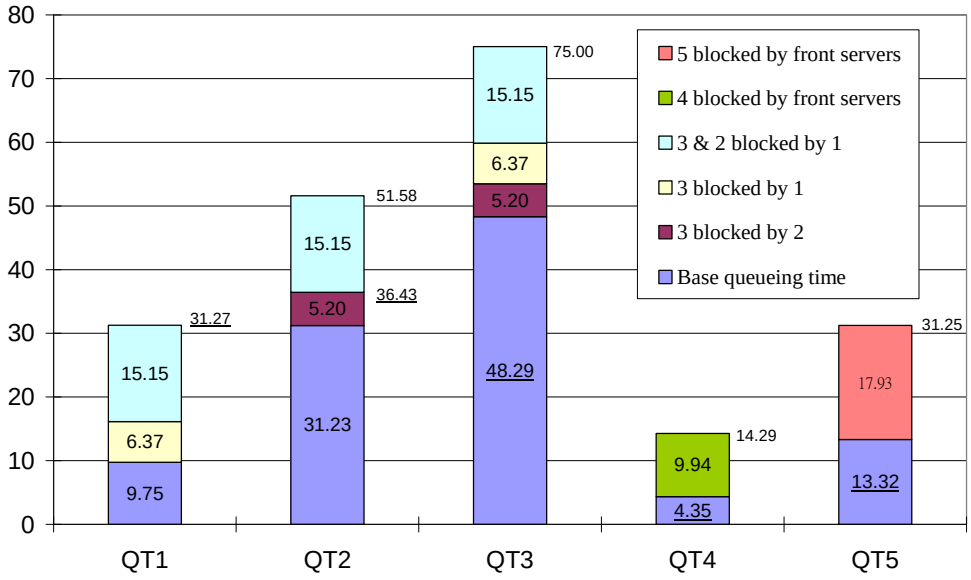

Figure 6.16 Queueing time analysis of the original system in Case B-2

Similar analysis can be done for the other traffic intensity of Case B-2 or the other cases. Since x_i (or y_i) decrease slowly with utilization increase as depicted in Figure 6.5, f_i and the ratio in Figure 6.16 should also possess the similar property.

Although, in Case B-2, the initial arrival process is Poisson, all other internal arrival processes are not Poisson and, indeed, are non-renewal. However, the blocking effect still exists and has nothing to do with renewal or non-renewal arrival processes. The blocking effect exists whenever the actual (or simulated) queueing time is shorter than its ASIA system queueing time.

6.6.2 Diffusion Effect

The true (or simulated) queueing time could be longer than its ASIA system queueing time. In this case, a "diffusion effect" occurs. A similar analysis to the blocking effect can be done. However, the value of f_i is greater than 1.

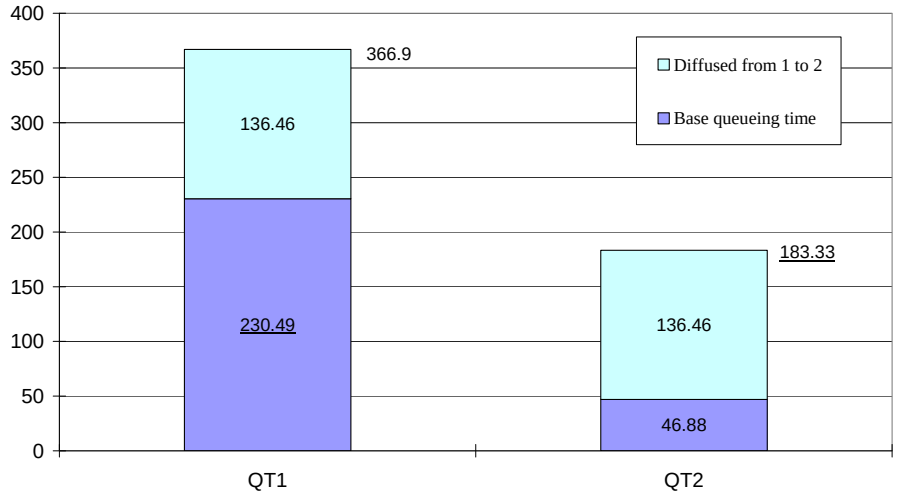

Figure 6.17 Queueing time analysis of the original system in Case B-6

We explain diffusion effects by analyzing the example given in Case B-6 when the arrival rate is 33.3 (i.e. 90% system utilization). The first sub-system in Case B-6 is composed of the first two servers. Their service times are 23 and 25 and service time SCVs are 8 and 0.25. Based on the simulation results in Table 6.14, the queueing times are 230.49 and 183.33 (the numbers with underlines in Figure 6.17). Since y_2 is 1.592, f_1 is 1.592 and f_2 is 1. The ASIA system queueing time of the second server is 46.88. Therefore, the contributions from the first and second server to the system queueing time are 366.95 and 46.88.

If the second server sees the initial arrival process, the queueing time of the second server (i.e. the bottleneck in this sub-system) should be 46.88. However, it is 183.33 when it sees the output of the first. The difference (of 136.46) between its true

queueing time (i.e. 183.33) and its independent queueing time (i.e. 46.88) is diffused from the first server in Figure 6.17.

If we continue analyzing the larger sub-systems one by one, we obtain the correlation among workstations in Case B-6, which are shown in Figure 6.18.

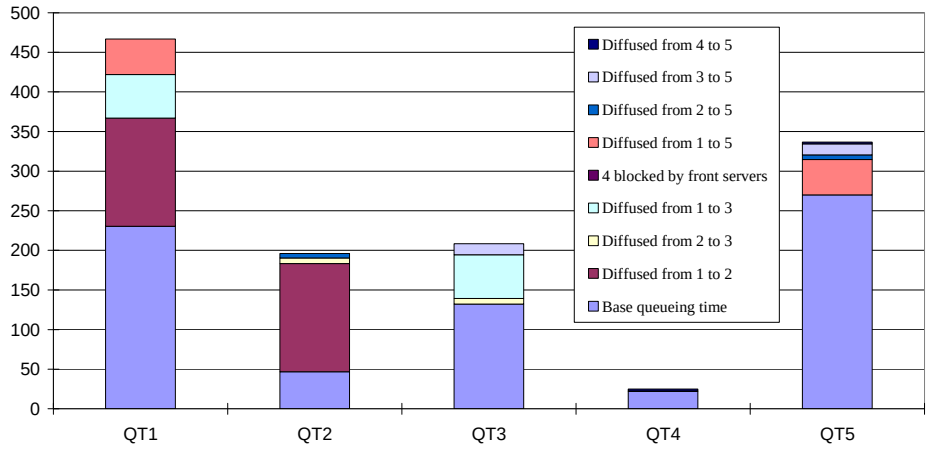

Figure 6.18 Queueing time analysis of the original system in Case B-6

The analysis of blocking and diffusion effects tells us the importance of the bottlenecks in manufacturing systems. Based on Marshall's equation, the bottleneck service time variability has more impacts on its downstream servers than those of the non-bottlenecks, which have lower utilizations. If a bottleneck currently generates diffusion effects, and we can reduce its service time variability, it could switch to inducing blocking effects on its downstream servers. The improvement on the system queueing time can be notable.

6.7 Conclusion

Using the concept of intrinsic ratio developed in Chapter 5, new approximate models for multiple single-server queues in series have been proposed. These new

approximations outperform QNA and QNET in most situations. Although they seem to perform well, additional research is needed. Further research is needed on the approximation of intrinsic ratios when service time SCVs are greater than 1 in STQF, when the arrival process is not Poisson, or when the one of the service time SCV in simple tandem queues is greater than the initial arrival process SCV and the other one is smaller then the initial arrival process SCV. Furthermore, multiple-server queues in series have not been studied.

Procedure 6.4 and 6.4a tell us how to approximate the queueing time of tandem queues using historical queueing time performance, without resorting to parametric-decomposition method. This practical approach considerably reduces the approximate errors. The blocking and diffusion effects give us a way to look at the correlation among general tandem queues. The analysis described in Section 6.6 can help us understand the behavior of tandem queues.

A tandem queue can be analyzed exactly in two special cases: Jackson networks and Freidman's reduction method. Identifying the common structure underlying Jackson networks and the fully coupled systems plays the key role in our new approach: both of them see the initial arrival process directly. This crucial insight gives us a key to open the door of general queueing networks. From this insight, we identified the ASIA system and the nice properties associated with the intrinsic gap. From the view point of ASIA systems, the behavior of tandem queues becomes more regular and predictable as we have seen in Section 6.5 and 6.6.

In practical manufacturing systems, in order to maintain competitiveness, service time SCV is desired to be small and system cycle time is required to be short while satisfying customer demand. If the service time SCV is small, the parametric-decomposition approach will not perform well. Since cycle time is required to be short and some machine costs are less than others, not all servers are in heavy traffic. When service time SCV is small and not all servers are in heavy traffic, Brownian motion based

model will not perform well in general manufacturing systems. The intrinsic ratio based approach gives a good alternative under practical situations.

In the next chapter, we extend the results on the present chapter to approximate the performance of manufacturing systems.

CHAPTER 7

CHARACTERIZING PERFORMANCES OF MANUFACTURING SYSTEMS

It is wrong to think that the task of physics is to find out how Nature is. Physics concerns what we say about Nature. ~ Niels Bohr

7.1 Introduction

Due to the complexity of manufacturing systems, it is difficult to describe the behavior exactly (considering batching, assembly lines, interruptions, reentry, product mixes, and dispatching rules, etc.). Instead of finding out how a manufacturing system behaves by considering every possible detail, it might be more practical to model a manufacturing system from a macroscopic view. Without describing the exact behavior of each particle, quantum mechanics describes the behavior of particles through the concept of energy levels. Similar to the role of energy levels in quantum mechanics, the underlying structure of tandem queues plays an important role in understanding the behavior of manufacturing systems.

Optimizing performance is an essential objective in factories. Quantifying the performance is the first step. Variability plays a key role in the process, since it can be used to characterize the trade-off between queueing time and utilization, as illustrated in Figure 7.1. Because the curve in Figure 7.1 characterizes the performance of a manufacturing system, it is sometimes called a performance curve, characteristic curve, trade-off curve, operation curve or queueing curve. Although it has many different names, they all refer to the same curve. Bitran and Tirupati (1989b) introduced the concept of performance curve to describe the relationship between WIP, cycle time and capacity. Sattler (1996) used performance curves to determine productivity improvements of a fab.

She assumed variability is independent of utilization and approximated the performance curve by using a constant k to replace the variability term in Kingman's approximation. Boebel and Ruelle (1996) used the same concept to measure the productivity improvement of the cycle time reduction program at the SIEMENS/IBM Advanced CMOS Line. Fowler et al. (1997) used a performance curve to measure the improvements in cycle-time-constrained capacity. Collins et al. (1997), Ruelle (1997), and Rose (2001) all used this curve to quantify the productivity improvement of a fab.

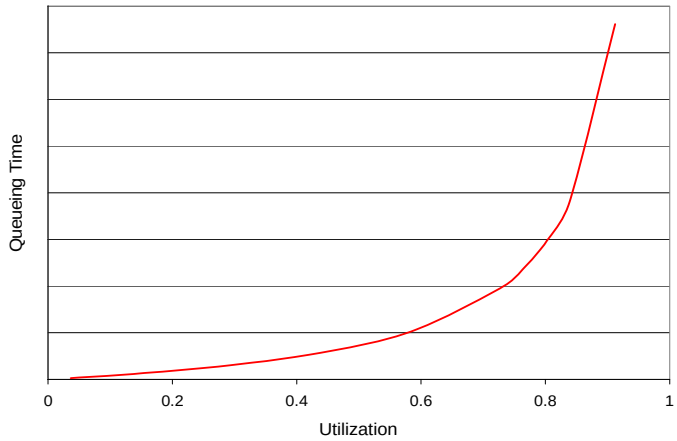

Figure 7.1 A performance curve showing queueing time versus utilization. Longer queueing time is the price of higher utilization.

Performance curves are commonly used to quantify the trade-off between cycle time and throughput. One way to generate performance curve is to use an analytical model. The challenge is find one which can describe the behavior of complex manufacturing systems accurately. From this view point, the study can be traced back to Chandy, Herzog and Woo (1975) who applied parametric analysis to queueing networks. They constructed an equivalent network in which all queues except those in a subsystem were replaced by a single composite queue. The behavior of the subsystem was then studied through the equivalent network. The subsystem in the equivalent network was

also called the flow equivalent server (FES). The authors showed that the behavior of the subsystem in their constructed equivalent network is the same as in the given network, for a limited class of systems.

Variability for a single machine system has been discussed in Part I. It can be expected that examination of variability for a manufacturing system will be much harder, since a manufacturing system is composed of many workstations, and the behavior of one workstation may depend on the behavior of others. Obtaining the exact performance curve analytically would be difficult in general.

Rather than creating the curve analytically, Rose (2001), Nazzal and Mollaghasemi (2001) and Park et al. (2001) developed the fab performance curve through simulations. Apart from simulation studies, Sattler (1996) and Collins et al. (1997) attempted to describe the performance curve of a fab based on Kingman's G/G/1 approximate model and demonstrated how to enhance productivity through variability reduction. However, rather than a single server, a fab is actually composed of a sequence of operations executed by a series of workstations. Since fab variability is the gross effect of the process time variability, flow variability and interactions among workstations, predicting fab performance curve simply based on the G/G/1 queue may not be adequate for practical manufacturing systems.

Based on the parametric-decomposition approach, Wu (2005) used a single machine system to gauge system variability, and subsequently derived an explicit expression for the variability of a simple factory. The simple factory is composed of single servers in series. In the analysis, the simple factory is decomposed into individual workstations, and each workstation has its specific SCV of service time and inter-arrival time. Since the workstations are assumed to be stochastically independent, the result from the decomposition approach is only an approximation. In that paper, the properties of variability for a simple factory are examined in terms of utilization versus throughput bottlenecks and non-throughput bottlenecks, gap effects, and bounds on variability.

Motivated by Kingman's approximation, Yang, Ankenman and Nelson (2007) proposed empirical algorithms to generate performance curves through simulation and metamodeling. Two unknown scalars and one unknown vector have to be determined through complex procedures. The algorithm performs well for M/M/1 systems with different dispatching rules. The algorithm is extended to predict the performance curve of manufacturing systems by ignoring the dependence among workstations.

The situation in a practical factory is much more complicated than a simple network model. In practice, there can be reentry, rework, batches, shift schedules, and multiple products with different priorities, etc., and each workstation can consist of multiple servers, where each server may have unique capabilities. It is difficult to describe the variability of a practical factory exactly. In order to have a good approximation, we must capture the important structure into the approximate model.

There are two approaches based on the availability of information. We will call them the "white box" and "black box" approach. The white box approach assumes the information (such as service time SCV or mean queueing time) of each machine is available and has been introduced in Chapter 6.

The "black box" approach treats the whole factory as a black box, where only system throughput rate and cycle time are known. The detailed information of each machine is assumed to be unknown. The objective in the black box approach is to find a model which can describe factory behavior through regression analysis. In other words, we use the known throughput rates and queueing times to fit the parameters of the model. After the parameters are determined, we can use the model to predict cycle times at other utilization levels. The derivation of the black box approach is given in Section 7.2. Model verification is given in Section 7.3. Conclusions are given in Section 7.4.

7.2 Performance of Manufacturing Systems with Single-Server Bottlenecks

A practical factory is much more complex than many single-server queues in series. It may have long process flow sequences with reentry and rework, each workstation may be composed of multiple servers with different capabilities and both random queueing time and asynchronous queueing time may exist. Each server may have complex configuration and suffer different types of interruptions as discussed in Chapter 2. Complex dispatching rules other than FCFS may be applied to each workstation. Under these conditions, understanding the behavior of a factory may not be an easy task.

However, if we want to optimize factory performance, describing the behavior of a factory quantitatively is essential. Rather than analyzing all activities in detail, we derive an approximate model by capturing the main underlying structure of a factory. In Chapter 6, the total queueing time of many queues in series can be described as

$$\sum_{i=1}^{n} QT_i = f_1\alpha_1\left(\frac{\rho_1}{1-\rho_1}\right)\frac{1}{\mu_1} + f_2\alpha_2\left(\frac{\rho_2}{1-\rho_2}\right)\frac{1}{\mu_2} + \ldots + f_n\alpha_n\left(\frac{\rho_n}{1-\rho_n}\right)\frac{1}{\mu_n}, \qquad (6.26)$$

where f_i is the contribution factor and can be approximated by a function of $(\lambda, c_{a1}^2, ST_i, c_{ei}^2)$ for i = 1 to n. Since f_{BN} is 1, cycle time is

$$CT = \sum_{i=1}^{n} QT_i + PT_f = \sum_{i=1}^{n} f_i\alpha_i\left(\frac{\rho_i}{1-\rho_i}\right)\frac{1}{\mu_i} + PT_f$$

$$= \alpha_{BN}\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + \sum_{i\neq BN} f_i\alpha_i\left(\frac{\rho_i}{1-\rho_i}\right)\frac{1}{\mu_i} + PT_f, \qquad (7.1)$$

where PT_f is total processing time, which is the minimum time that a job needs to complete its process. When no batching or assembly exists, processing time is the cycle time of a job in light traffic. The first term of Eq. (7.1) is the ASIA system queueing time of the bottleneck, and the second term is the gross queueing time of the non-bottlenecks.

Variability of tandem queues is determined by all servers. The coefficient (α_{BN}) of the bottleneck is the same as the variability in its ASIA system. The coefficients of the

non-bottlenecks are the weighted (by their contribution factors) variabilities ($f_i \alpha_i$) in their ASIA systems. When all service times are exponential and the initial arrival process is Poisson, f_i is 1 and Eq. (7.1) reduces to the cycle time of a Jackson network. If all service times are deterministic, f_i is 0 and Eq. (7.1) reduces to a fully coupled system.

In order to apply Eq. (7.1), we assume the SCV of service times are available. However, we need to keep in mind that mean service time, by definition, is the reciprocal of capacity, where capacity is the maximum throughput rate. In a practical production environment, service time usually is not the same as processing time.

If we ask a planner, "What is the variance of service time, where its mean is the reciprocal of machine capacity?" rather than finding the answer, he may first ask us what we are really asking about. As we mentioned in Chapter 4, in a setting of complex machine configurations, robot scheduling, interruptions, resource contention, reentry and product mix, finding out the variance of service times is not trivial. Newell (1979) stated, "In fact, in most applications, one is lucky if one has a good estimate of the service rates (to within 5% say); the variance rates are often known only to within a factor of 2, seldom to within an accuracy of 20%." Although SCV of service time is well defined in theory, it may not be so accessible in practice.

One way to get a reliable estimate of the service time SCV is to analyze the historical data directly, but even that may not be simple. If we have to analyze the historical data, we may estimate other parameters at the same time as well, such as mean queueing time. In practice, observable mean queueing time (first moment estimator) is much more accessible than the intangible service times SCV (second moment estimator).

In order to make queueing models more accessible to practitioners, it would be nice to have a new approach which does not rely on the service time SCV explicitly, but takes it into account implicitly. The historical queueing time is a good alternative. Indeed, except for constructing a brand new factory, in practical applications of queueing theory, we usually know the historical factory performance but want to know the performance in

the future at different utilizations. Especially in semiconductor fabs, since the product cycle time can be one or two months (or even longer) and customer demand may last for a while, product mix usually changes gradually rather than dramatically. In this situation, the approximate models may give us a good approximation of system performance for the near term. Therefore, we want to develop an approximate model which can predict future factory performance simply based on historical queueing times.

Figure 7.2 gives a graphical demonstration of the queueing times for five single M/M/1 queues in tandem. Their service times are 20, 23, 25, 27, and 30. It shows that the performance curves of the four non-bottlenecks (with service times from 20 ~ 27) are much closer to each other, compared to the bottleneck performance curve (i.e. solid line).

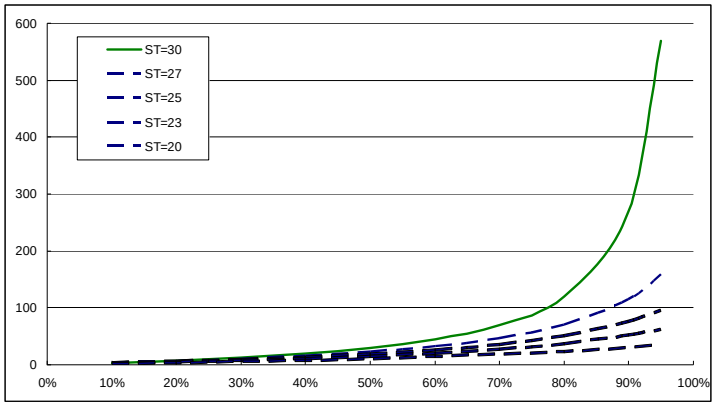

Figure 7.2 Queueing times of five M/M/1 queues

When service time SCVs are smaller than 1 and the initial arrival process is Poisson, f_i is smaller than 1. The difference between the bottleneck workstation performance curve and non-bottleneck ones will become even larger. Therefore, if we want to further simplify the model, it is possible to replace all non-bottleneck performance curves by four identical composite non-bottleneck performance curves.

Although the non-bottleneck performance curves are very close to each other in light traffic, they have considerable differences in heavy traffic. Since, within all non-

bottleneck performance curves, system cycle time is dominated by the performance curve of the second bottleneck (i.e. the one with the highest utilization among all non-bottlenecks) in heavy traffic, a reasonable choice is to pick up the performance curve of second system bottleneck to represent the composite non-bottleneck performance curves.

Based on the above observations, Eq. (7.1) can be simplified as

$$\begin{aligned} CT &= \alpha_{BN}\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + \sum_{i\neq BN} f_i\alpha_i\left(\frac{\rho_i}{1-\rho_i}\right)\frac{1}{\mu_i} + PT_f \\ &\cong k_1\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + (n-1)k_2'\left(\frac{\lambda/k_3}{1-\lambda/k_3}\right)\frac{1}{k_3} + PT_f \\ &= k_1\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{\mu_{BN}} + k_2\left(\frac{\lambda}{k_3-\lambda}\right)\frac{1}{k_3} + PT_f, \end{aligned} \tag{7.2}$$

In this model, the first term can be interpreted as corresponding to the bottleneck queuing time with k_1 as the bottleneck variability. The second term can be interpreted as corresponding to queuing time at a composite non-bottleneck station, with the constant k_2 approximating the variability of this composite station (representing the (n – 1) non-bottleneck stations), and k_3 representing the composite non-bottleneck capacity. It should be noted that k_1 is the same as α_{BN}, the bottleneck variability in the ASIA system, if the system is a tandem queue without feedbacks.

When there is reentry or rework, capacity is the reciprocal of the summation of all service times weighted by the rework rate. Therefore,

$$1/\mu_i = \sum_{j=1}^{l} w_j \times ST_j,$$

where l is the total reentry and rework frequency at station i, w_j is the rework rate when ST_j is the length of a rework (and w_j is 1 when it is the length of a reentry). Since there are three parameters in Eq. (7.2), we call it the 3-parameter model.

Although Eq. (7.2) is motivated by the underlying structure of tandem queues, as we will see later, it performs very well for the practical manufacturing systems examined, even with reentry and rework. Therefore, we have the following conjecture.

Conjecture 7.1 (Behavior of Manufacturing Systems):

Behavior of a stochastic manufacturing system is dominated by the underlying structure observed from simple tandem queues.

Conjecture 7.1 will be tested in Section 7.3 using real cases from industry. When applying Eq. (7.2) to a specific factory, the values of k_1, k_2 and k_3 should be determined considering practical issues such as reentry, dispatching rules and interruptions. Obviously, calculating k_1, k_2 and k_3 analytically is difficult. One way to determine their values is by multiple regression analysis, if the historical performance curve is available. Then, factory variability can be approximated by k_1 and k_2. *We say the performance of a factory is improved, if the value of k_1 or k_2 becomes smaller at a given traffic intensity.* The parameters k_1 and k_2 describe the variability of a factory in the approximate model of Eq. (7.2). Therefore, considering both Eq. (7.2), it may be concluded that factory variability can be lowered by reducing the service time variability, the initial arrival process variability, or the number of non-bottlenecks.

Based on Procedure 6.2, if there are multiple bottlenecks (i.e. more than one server, which has the same highest utilization), only the one with the smallest sequence number is marked as the bottleneck. Similar to the approach proposed by Wu (2005), Eq. (7.2) gauges the variability of a manufacturing system from the viewpoint of the bottleneck, but adding a correction term to consider the impact from non-bottlenecks.

7.2.1 An Industrial Case

The explanation of factory variability is based on an important assumption: Eq. (7.2) can describe factory performance curves accurately. In this section, we will test this assumption with a real manufacturing system case.

Planning and managing major defense acquisition programs (DAP) requires balancing and synchronizing design and production across a network of distributed activities performed by independent commercial entities. We need to have the capability to accurately describe the performance curve of each entity.

To achieve this goal, a simulation model is constructed by one of the manufacturers in a DAP using ARENA®. The model (illustrated in Figure 7.3) describes the behavior of their manufacturing facility for a specific product.

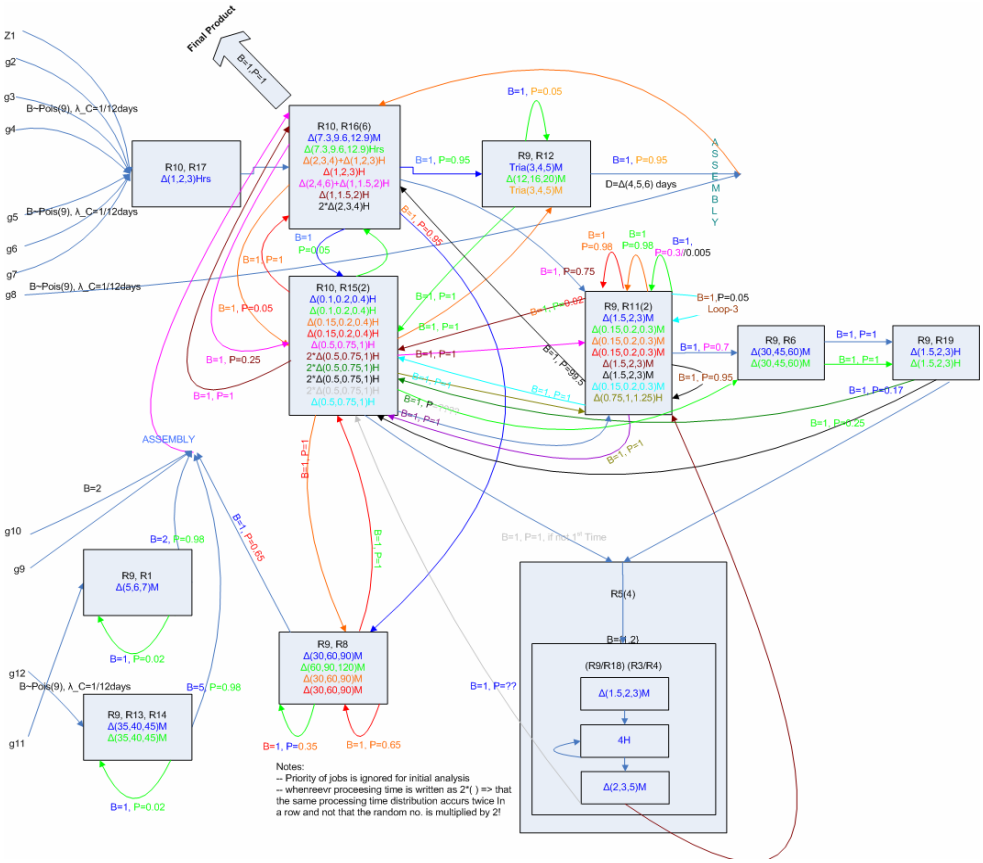

Figure 7.3 Process flows of a manufacturing facility in DAP

The output of this facility supplies critical parts to downstream assembly lines in the supply chain. There are 14 workstations arranged in 11 main process groups (i.e. 9 process groups use 1 workstation, 1 group needs 2 workstations, and 1 group uses 3

workstations). Four workstations have multiple servers (i.e. R5=4, R11=2, R15=2, R16=6), and all the rest have only a single server. In addition to workstations, each process group requires operators of one of the two operator types (R9 and R10). While the operators have their own shift schedules (i.e. 8 hours a day), machines work 24 hours a day as long as a job has been loaded by operators.

In the factory simulation model, service times follow triangular distributions and dispatching rules at some critical workstations are the shortest remaining processing time instead of FCFS. Reentry and rework are observed in the system as shown in Figure 7.3. 12 different raw parts arrive every 12 days with random batch size following a Poisson distribution. System utilization is determined by the mean batch size rather than the arrival intervals. Before the process can be started, raw parts (e.g. Z1 and g2 ~ g7) need to be assembled in front of the first process step. Afterwards, an incomplete job has to be assembled again with some other raw parts (e.g. g8 ~ g12) in the middle of the process flow. The cycle time of a job is the duration between its process start and departure.

Although this model seems complex enough, it is still simpler than the situation in a semiconductor fab. The machines in semiconductor fabs usually possess complex configurations. Cluster tools or multiple chambers are commonly seen. Different chambers or tools are connected by robots with intricate scheduling rules. Chambers, tools and robots are subject to interruptions. Not all these condition exist in the simulation model of this manufacturing facility.

Although the model is relatively simple, finding good estimations of inter-arrival time and service time SCVs is not simple at all, considering the shift schedule, operator availability, batch arrivals and assembly lines. Therefore, even finding a reliable cycle time approximation for this system is difficult. It is reasonable to resort to simulation. Through experimentation, the system bottleneck has been identified to be R8, which is composed of a single server, and system capacity estimated as 13.7 jobs per 12 days (indeed, even finding the bottleneck and true capacity is not trivial). In total, fourteen

different utilization levels have been simulated. The performance curve with 99% confidence interval is shown in Figure 7.4. Each cycle time value is the average of 31 batches from one long simulation run. Depending on the model output variance, each batch consists of 1,500 ~ 500,000 data points. In each simulation run, the first 1,000 ~ 10,000 data points are discarded for warm-up depending on the utilization levels.

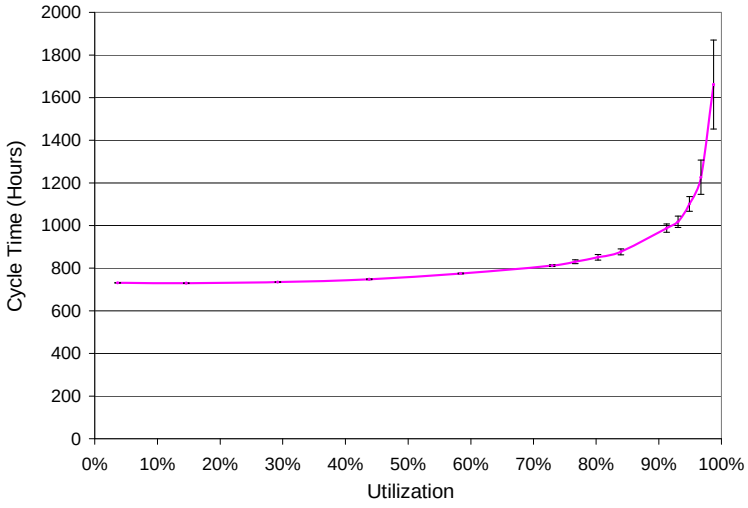

Figure 7.4 Performance curve of a manufacturing facility in DAP

Sattler (1996), Collins et al. (1997) and Wu (2005) attempted to describe the performance curve based on Kingman's approximation,

$$CT = k_1 \left(\frac{\rho_{BN}}{1-\rho_{BN}} \right) \frac{1}{\mu_{BN}} + PT_f . \qquad (7.3)$$

Since there is only a single parameter in Eq. (7.3), we call it the single-parameter model. Based on reduction method, Eq. (7.3) describes system performance curve exactly when all service times are deterministic, and approximately otherwise. It is important to compare the performance of single- and three-parameter models and to see how much improvement the three-parameter model can make.

Furthermore, motivated by Kingman's queueing time approximation, Yang, Ankenman and Nelson (2007) proposed the following model to describe the cycle time of manufacturing systems:

$$CT(x, \boldsymbol{c}, p) = \frac{\sum_{k=0}^{t} c_k x^k}{(1-x)^p}, \tag{7.4}$$

where x is system throughput rate. p, t and the vector $\boldsymbol{c}$ = (c_0, c_1, …, c_t) are unknown parameters in the model. We call Eq. (7.4) YAN's model (for the initials of the three authors). Since t can increase without limit, we limit the value of t to be 2 in order to have a fair comparison with Eq. (7.2). Eq. (7.4) becomes

$$CT(x, \boldsymbol{c}, p) = \frac{c_0 + c_1 x + c_2 x^2}{(1-x)^p}. \tag{7.5}$$

There are 3 terms in Eq. (7.5), which is the same as Eq. (7.2), and 4 parameters (p, c_0, c_1, c_2), which is one more than Eq. (7.2).

In order to compare the performance of the three models, the simulated cycle time in Figure 7.4 is used to fit the parameters in Eq. (7.2), (7.3) and (7.5) by the statistical software package, R.

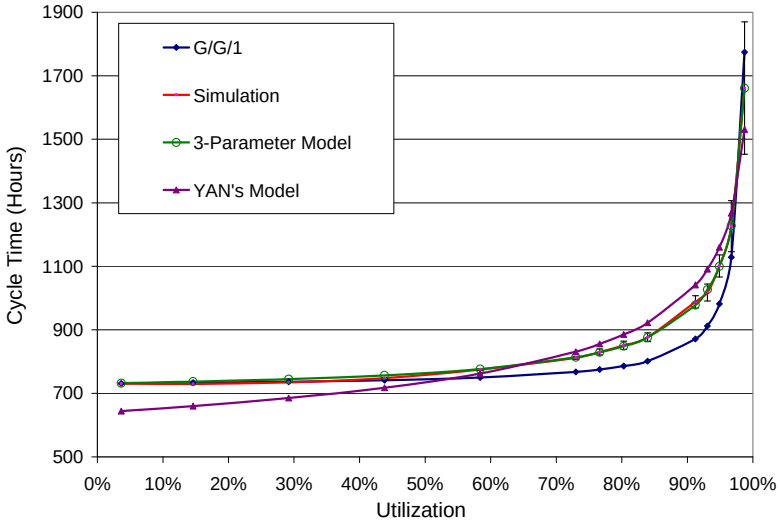

Figure 7.5 Fitting results of different models

The value of k_1 in Eq. (7.3) (i.e. the G/G/1 curve in Figure 7.5) is 0.642, and the largest error is 11.77% at 91% utilization. The values of c_0, c_1, c_2 and p in Eq. (7.5) (i.e. the YAN's curve in Figure 7.5) are 639.452, 0, 0 and 0.200, respectively, and the largest error is 11.87% at 4% utilization. The values of k_1, k_2 and k_3 in Eq. (7.2) (i.e. the 3-Parameter fitting curve in Figure 7.5) are 0.356, 422.951 and 14.636, and the largest error is 1.38% at 29% utilization. It should be noted the value of k_3, 14.636, which represents the non-bottleneck capacity, is greater than the bottleneck capacity 13.7 as expected.

In Figure 7.6, the three-parameter model gives the best fit among the three. Furthermore, the three-parameter model performs very well for utilization of 60 ~ 85% which represents a realistic case in practice. YAN's model gives large errors at low utilization, since Eq. (7.4) is inspired by queueing time instead of cycle time models. The ignored processing time would be responsible for those errors.

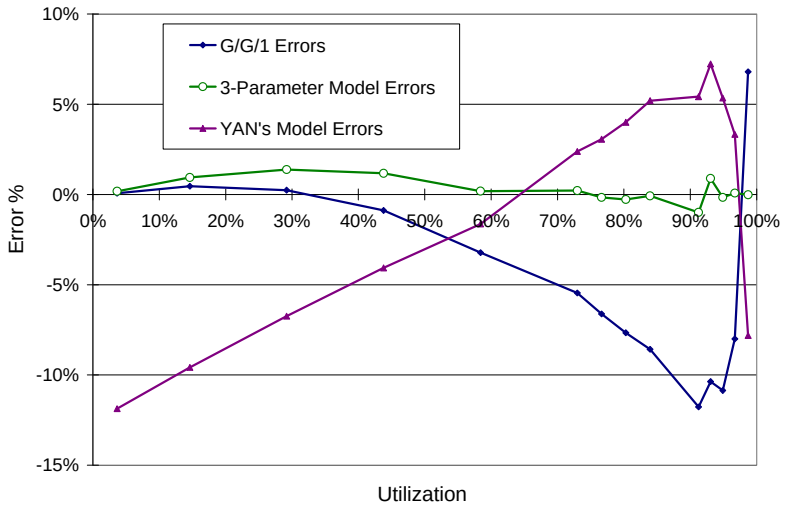

Figure 7.6 Fitting errors of different models

This simulation experiment exhibits a strong prediction capability for Eq. (7.2). The question becomes how to approximate k_1, k_2 and k_3 accurately, since we would not have so many historical data points in practice. The insight on the three parameters, and

the method described in Section 7.2.1 becomes important. We are going to describe and demonstrate a procedure in the next example.

7.3 Implementation in manufacturing Systems with Single-Server Bottlenecks

When the performance curve is known, k_1, k_2 and k_3 in Eq. (7.2) can be obtained by regression analysis. This requirement could be satisfied if the simulation model is available, but may not be realistic in practice.

In practice, steady states may never be attained. Furthermore, queueing time has large variations (especially in heavy traffic) even in steady state. When approximating queueing time of practical manufacturing systems, a large sample size from a long observation period is necessary for one single approximation of queueing time at a specific utilization level (e.g. planned monthly production rate / capacity). It would not be realistic to expect the complete performance curve to be available in practice.

This issue can be resolved by using our intuitive interpretation of the terms in Eq. (7.2), along with specific information about the first and second bottlenecks in the system. If the bottleneck queueing time in its ASIA system is known, since f_{BN} is always 1, we can estimate k_1 directly, using only the first term of Eq. (7.2). If we let the second bottleneck represent all of the non-bottleneck stations, then k_3 can be approximated by the capacity of the second bottleneck, i.e.

$$k_3 = \min \mu_i, \ \ i \neq BN .$$

After k_1 and k_3 are known, k_2 can be determined by regressing the historical factory cycle time against Eq. (7.2) with the values of k_1 and k_3 specified. In this approach, a one point estimator of the historical factory and bottleneck queueing time suffices. Furthermore, if multiple data points are available, the previous multiple regression analysis can be reduced to single regression analysis for k_2 only. Therefore, we call the above method the k2 regression model (or, in short, the k2 model).

One nice feature of the k2 regression model is that we can approximate system queueing times without knowing the service time SCV, which is hard to obtain in practical manufacturing systems.

The challenge of the above approach would come from the estimation of the bottleneck queueing time in the ASIA system. Since the intrinsic gap ratio at the bottleneck goes to zero in heavy traffic, the bottleneck queueing time approaches its ASIA system queueing time in heavy traffic. This is consistent with the heavy-traffic bottleneck phenomenon observed by Iglehart and Whitt (1970): the queueing time distribution at the bottleneck is asymptotically the same as if the immediate arrival process were replaced by the external initial arrival process to the first queue. Since this condition is the same as our requirement for the ASIA system, we could approximate k_1 by using the historical bottleneck queueing time in heavy traffic.

Based on this approach, we can predict factory performance based on the historical queueing times without using the information of service time variability.

7.3.1 Flow Shop Simulator Example

DSN Innovations (or just DSN) is a "non-profit organization focused on bolstering U.S. manufacturing through research and innovations designed to improve manufacturing supplier network coordination, agility and efficiency" (DNS, 2009). One of the tools they have developed is a flow shop simulator that can be quickly populated with data from a small manufacturer and used to evaluate WIP at each workstation and overall manufacturing cycle time. Although the kernel of the tool is ARENA®, it uses Excel as its data input interface, so users don't require working knowledge of ARENA®.

DNS provided a case based on a manufacturing system consisting of five workstations as shown in Figure 7.7. While station 1 and 5 are visited only once, station 2, 3 and 4 are visited multiple times. Station 2, 3 and 4 can execute multiple job functions:

station 2 can do two different recipes, station 3 can do three and station 4 can do seven. There are total 14 process steps. There is a constant 20 minutes delay between the first and the second steps. Some steps may need to be reworked if the finished jobs are out of spec. All stations are composed of one single machine except for station 5, which contains 24 machines in parallel. The initial arrival process is Poisson. All dispatching rules are FCFS. The service time distribution is triangular and the data of each process step is shown in Table 7.1. Based on Table 7.1 and Figure 7.7, the capacities of station 1 to 5 are 62.609, 51.429, 49.655, 49.021 and 54.857 jobs/day and service times are 23, 28, 29, 29.375 and 630 minutes, respectively. The forth station is the bottleneck.

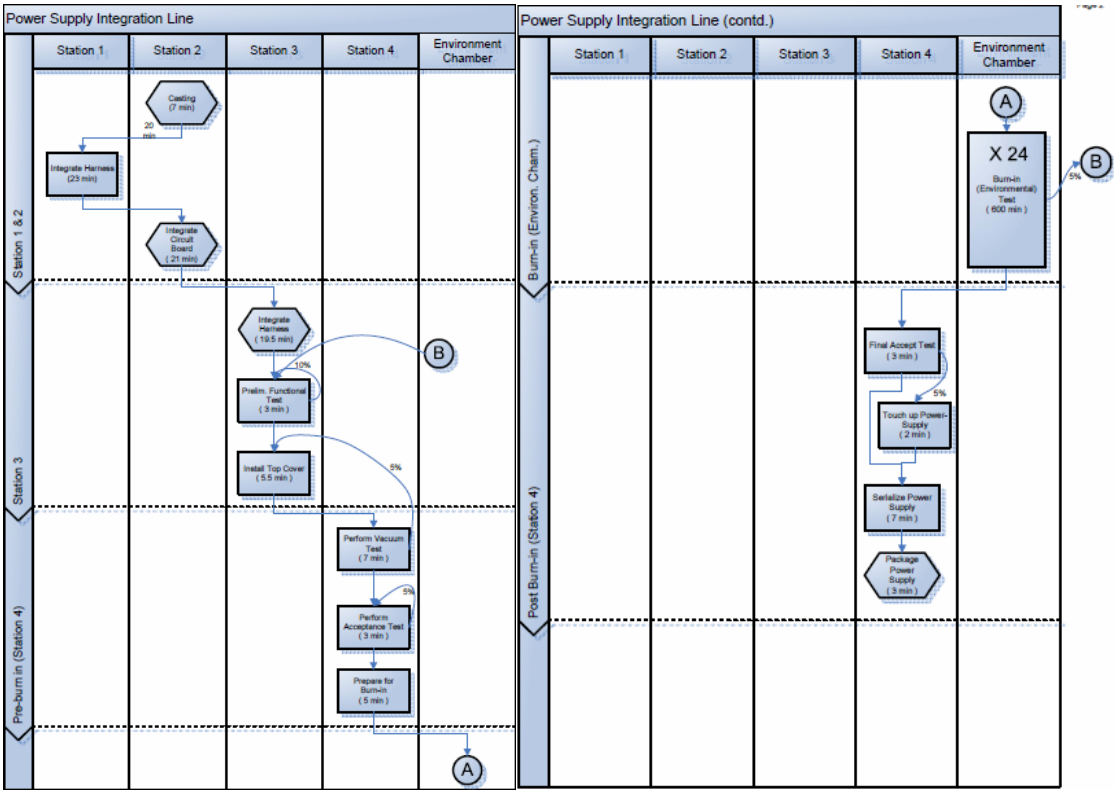

Figure 7.7 Process flow of the DNS model

Table 7.1 Process times of each process step

Process Number	Process Description	Process Resource	Processing Times (min.)		
			Min	Mode	Max
1	Bond Overlay and O-Ring	Station 2	6	7	8
2	Integrate Harness at Station 1	Station 1	22	23	24
3	Integrate Circuit Boards at Station 2	Station 2	20	21	22
4	Integrate Harness at Station 3	Station 3	18.5	19.5	20.5
5	Perform Preliminary Func Test	Station 3	2	3	4
6	Install Top Cover	Station 3	4.5	5.5	6.5
7	Perform Vacuum Test	Station 4	6	7	8
8	Perform Acceptance Test	Station 4	2	3	4
9	Prepare for Burn-in Test	Station 4	4	5	6
10	Burn-in (Environmental) Test	Environ. Chamber	600	600	600
11	Perform Final Accept Test	Station 4	2	3	4
12	Touch-up Power Supply	Station 4	1	2	3
13	Serialize Power Supply	Station 4	7	7	7
14	Package Power Supply	Station 4	3	3	3

Although this is a small scale manufacturing system, because of the complex reentry and rework, as well as the small service time variability, it is not easy to get reliable cycle time approximations by conventional queuing theory approaches. Thus, up to now the only viable approach would be simulation. The simulated cycle times and half-width 95% confidence interval at a number of input rates are shown in Table 7.2. Each observation in Table 7.2 is the average of 100 replications. Each replication is the collection of the output data in 10,000 days after a warm-up period of 100 years. The total system process time was estimated as 761.05 minutes by simulating a very low shop throughput.

Table 7.2 Cycle times from simulations and the historical data approach

Input Rate (per day)	Utilization	Simulation		k2 Model	
		CT (min)	95% CI	CT (min)	Error
4	8.2%	765.6	0.15	764.5	-0.15%
8	16.3%	771.0	0.10	768.5	-0.32%
12	24.5%	777.0	0.09	773.5	-0.45%
16	32.6%	784.2	0.08	779.6	-0.59%
20	40.8%	792.9	0.08	787.3	-0.70%
24	49.0%	803.7	0.08	797.5	-0.77%
28	57.1%	817.6	0.08	811.4	-0.76%
32	65.3%	837.2	0.13	831.7	-0.66%
36	73.4%	867.5	0.18	863.8	-0.43%
40	81.6%	922.7	0.38	922.6	0.00%
44	89.8%	1054.9	1.03	1065.2	0.97%
46	93.8%	1245.1	3.18	1254.6	0.76%

7.3.1.1 Historical Data Approach

Since this simulation model is simpler than the one presented in section 7.2, we can get the bottleneck queueing time and the non-bottleneck capacity relatively easier. Therefore, we can demonstrate the historical data approach by using the simulated data as our "historical" data. We use the system queueing time at 81.6% utilization and the bottleneck queueing time at 93.8% utilization (in heavy traffic). Based on simulations, the system queueing time is 161.63 minutes and the bottleneck queueing time is 23.046 minutes.

Because k_1 can be approximated by the variability of the bottleneck in heavy traffic, based on the first term of Eq. (7.2) and assuming the other two terms are zero, k_1 can be computed as 0.052. Since k_3 is the capacity of the second bottleneck, k_3 is 49.655 jobs/day (i.e. capacity of the third station). To estimate k_2, we have re-run the simulation model at 81.6% utilization with different random seeds. The cycle time is 922.65, which is used to represent the historical performance. Based on the values of k_1, k_3 and the system cycle time at 81.6% utilization, k_2 is 1857.412. Based on Eq. (7.2), cycle times at other utilizations can be calculated as shown in Table 7.2 (specified as the k2 model). The simulated and approximate cycle times are shown in Figure 7.8.

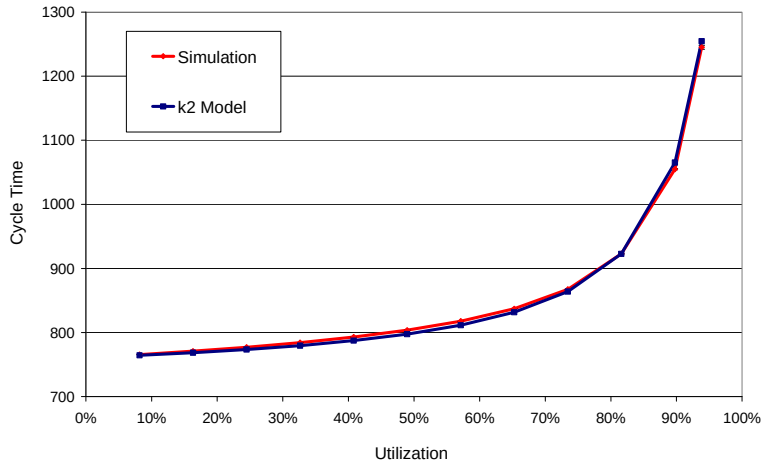

Figure 7.8 Performance curves of the Doyle Center Model

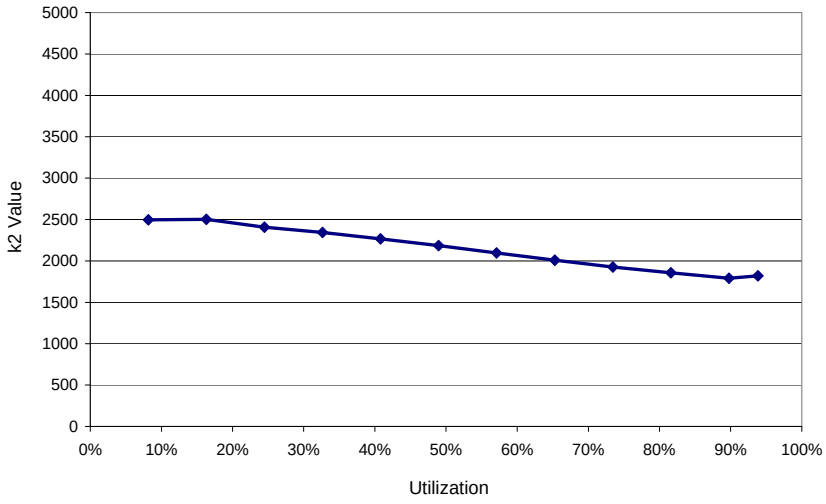

Figure 7.9 k_2 values at different utilizations

By using only few historical data points, we get a good approximation of the system cycle time based on the insight of the underlying structure. The biggest error among all utilizations is only 0.97% as shown in Table 7.2. Since we calculate k_2 based on the system cycle time at 81.6% utilization, the error is small at that specific utilization.

The result is attained by assuming k_2 is a constant across all utilizations. It would be interesting to verify this assumption by calculating the values of k_2 across all utilizations. The results are shown in Figure 7.9. The k_2 values all fall between 1790 and 2525, which explains the good approximate results in Figure 7.8. The regularly decreasing curve also suggests us that the approximation can be further improved by extrapolating the k_2 values if more than two queueing times are known at different utilizations (similar to what we have done in Procedure 6.4a).

Furthermore, one possible explanation for the declining curve is that the true value of k_3 is overestimated by the second bottleneck station capacity. For a given value of λ, the second term in the model is smaller than it should be, and the effect is exaggerated in heavy traffic.

7.3.1.2 Regression Analysis

In this section, the performances of the 3-parameter model, G/G/1 model and YAN's models are compared based on regression analysis. The simulated cycle time in Table 7.2 is used to fit the parameters in Eq. (7.2), (7.3) and (7.5) by the statistical software package, R. The results are shown in Figure 7.10.

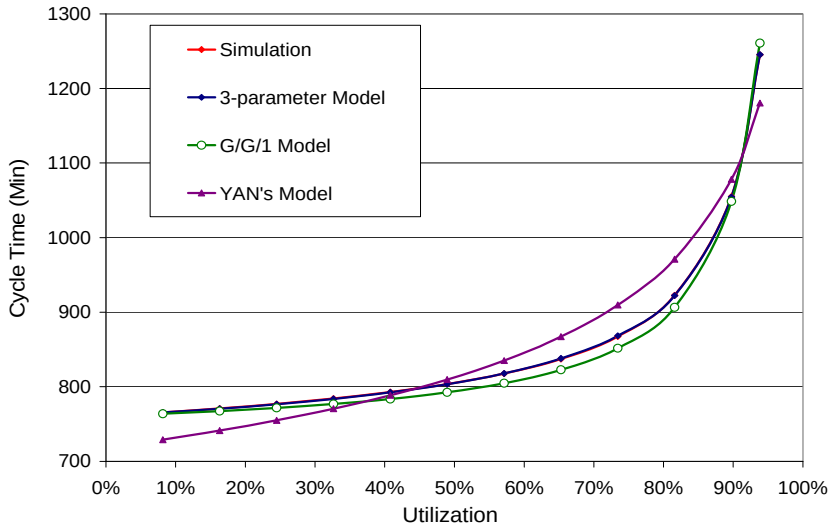

Figure 7.10 Fitting results of different models

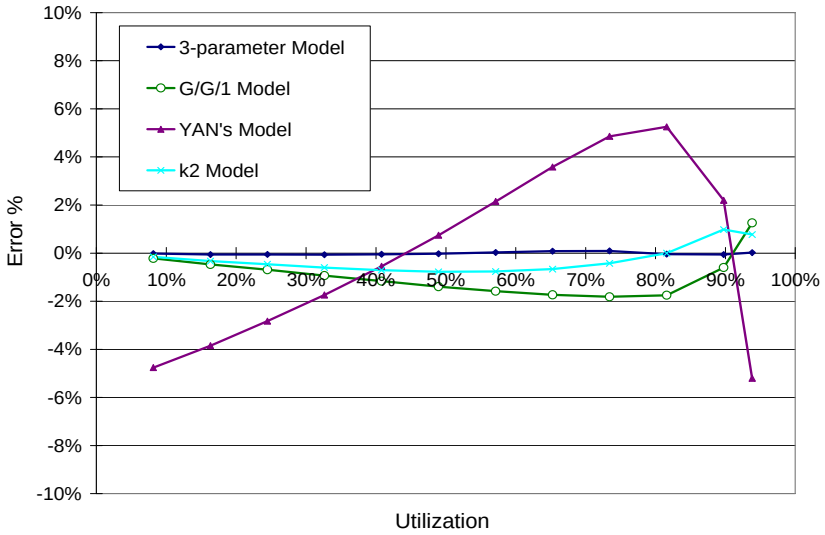

Figure 7.11 Fitting errors of different models

The value of k_1 in Eq. (7.3) (the G/G/1 model) is 1.117, and the largest error is -1.81% at 73.4% utilization. The values of c_0, c_1, c_2 and p in Eq. (7.5) (YAN's model) are 718.164, 0, 0 and 0.178, respectively, and the largest error is 5.26% at 81.6% utilization. The values of k_1, k_2 and k_3 in Eq. (7.2) (the 3-Parameter model) are 0.987, 3367 and 87.15, and the largest error is 0.10% at 73.4% utilization. The errors of all four models are shown in Figure 7.11.

In Figure 7.11, the three-parameter model gives the smallest errors. Furthermore, although the k2 model only uses one or two historical data points, it outperforms the fitted results of the G/G/1 and YAN's model, which use much more historical information. Because it needs limited historical data but offers high accuracy, the k2 model is suitable for use in practical manufacturing systems.

7.4 Performance of Manufacturing Systems with Multiple-Server Bottlenecks

Although some of the non-bottlenecks have multiple servers in the previous cases, since the bottleneck has only one single server, Eq. (7.2) still gives us good fitted results. In practical manufacturing systems, the bottleneck may not be composed of one single server. An extension of Eq. (7.2) to the multiple server cases is necessary.

When a workstation is composed of multiple servers, Kingman's approximation can be modified based on the M/M/m approximations proposed by Sakasegawa (1977) as follows,

$$E(QT_{G/G/m}) \cong \left(\frac{c_a^2 + c_s^2}{2}\right) E(QT_{M/M/m}), \tag{7.6}$$

where

$$E(QT_{M/M/m}) \cong \frac{\rho^{\sqrt{2(m+1)}-1}}{(1-\rho)} \frac{1}{m\mu},$$

and m is the number of servers of the workstation and $\rho = \lambda / (m\mu)$. Whitt (1993) further improved this model by adding correction factors.

Based on Eq. (7.6), Eq. (7.2) can be modified as follows,

$$CT \cong k_1\left(\frac{\rho_{BN}^{\sqrt{2(m_{BN}+1)}-1}}{(1-\rho_{BN})}\right)\frac{1}{m_{BN}\mu_{BN}} + k_2\frac{\left(\frac{\lambda}{m_2k_3}\right)^{\sqrt{2(m_2+1)}-1}}{\left(1-\frac{\lambda}{m_2k_3}\right)}\frac{1}{m_2k_3} + PT_f \qquad (7.7)$$

where m_{BN} is the number of servers at the bottleneck and m_2 is number of servers at the non-bottleneck. All the other parameters are the same as Eq. (7.2). If k_3 is the second bottleneck capacity, m_2 is the number of servers at the second bottleneck.

One potential issue of Eq. (7.7) is that the non-bottleneck cycle time may not be dominated by the second bottleneck, when its server numbers are considerably larger than the third (or forth, and so on) bottleneck workstations, except at extremely high utilization (which is almost unlikely, since they are not the system bottleneck). In this situation, the choice of k_3 may have to be adjusted. However, when server numbers do not differ too much among those non-bottlenecks with higher utilizations (e.g. the second to the fifth bottlenecks), choosing the second bottleneck should suffice.

From Eq. (7.7), system cycle time decreases when the server number increases. Therefore, Eq. (7.7) can be transformed as follows,

$$QT \cong k_1\rho_{BN}^{\sqrt{2(m_{BN}+1)}-2}\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{m_{BN}\mu_{BN}} + k_2\left(\frac{\lambda}{m_2k_3}\right)^{\sqrt{2(m_2+1)}-2}\frac{\frac{\lambda}{m_2k_3}}{\left(1-\frac{\lambda}{m_2k_3}\right)}\frac{1}{m_2k_3}$$

$$= \bar{k}_1\left(\frac{\rho_{BN}}{1-\rho_{BN}}\right)\frac{1}{m_{BN}\mu_{BN}} + \bar{k}_2\frac{\frac{\lambda}{m_2k_3}}{\left(1-\frac{\lambda}{m_2k_3}\right)}\frac{1}{m_2k_3}. \qquad (7.8)$$

where $\bar{k}_1 = k_1 \rho_{BN}^{\sqrt{2(m_{BN}+1)}-2}$, and $\bar{k}_2 = k_2 \left(\frac{\lambda}{m_2 k_3} \right)^{\sqrt{2(m_2+1)}-2}$.

In Eq. (7.8), variability ($\bar{k}_1$ or $\bar{k}_2$) depends on utilization when *m* is greater than one. At a given utilization, variability decreases when *m* increases.

7.4.1 Flow Shop Simulator Case with Multiple Servers

In order to verify Eq. (7.7), the original DSN model has been modified. The servers of all workstations (as well as the input rate) become 5 times more than the previous model. Therefore, all stations now are composed of 5 servers in parallel except for station 5, which contains 120 servers. The process times and process flows are the same as in the original case.

7.4.1.1 Historical Data Approach

From simulations, the bottleneck queueing time is 8.58 minutes at 94.8% utilization. Therefore, the k_1 value is approximated in a similar manner as in Section 7.3.1.1, yielding 0.087. The k_3 value can be approximated by the second bottleneck capacity, 248.276 (i.e. capacity of the third workstation). The total system process time is estimated to be 761.05 minutes from a very low throughput rate simulation.

By re-running the simulation at 89.8% utilization (with different random seeds) and assuming the outcome (i.e. 818.32) represents the historical cycle time, Eq. (7.7) can be solved for k_2 (4604.2) because all the other values are known. The approximated cycle times and corresponding errors at other utilizations are shown in Table 7.3 and Figure 7.12. The largest error is -2.16% at 94.8% utilization.

Table 7.3 Cycle times from simulations and the single point k2 model

Input Rate (per day)	Utilization	Simulation		k2 Model	
		CT (min)	95% CI	CT (min)	Error
20	8.2%	761.1	0.07	761.1	0.00%
40	16.3%	761.2	0.04	761.3	0.01%
60	24.5%	761.2	0.03	761.6	0.05%
80	32.6%	761.6	0.03	762.3	0.10%
100	40.8%	762.2	0.03	763.4	0.15%
120	49.0%	763.3	0.03	764.9	0.21%
140	57.1%	765.3	0.03	767.3	0.27%
160	65.3%	768.4	0.03	771.1	0.34%
180	73.4%	774.0	0.02	777.3	0.43%
200	81.6%	785.1	0.04	789.1	0.51%
220	89.8%	818.2	0.12	818.3	0.02%
232	94.8%	892.8	0.43	873.5	-2.16%

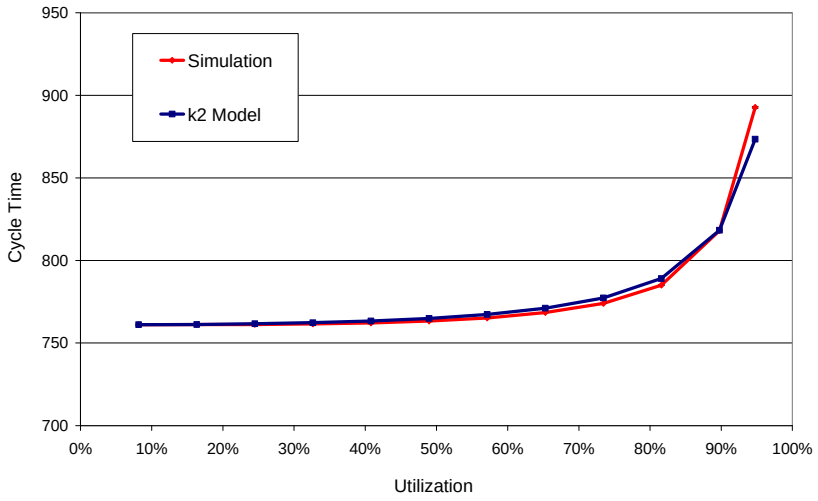

Figure 7.12 Performance curves based on single point k2 method

In Figure 7.13, the known k_1 and k_3 values and the simulated queuing times at different utilizations are used to estimate the corresponding k_2 values. The displayed values of k_2 seem to exhibit a very regular behavior, and this suggests that we may improve the results by using a two point approximation method. To illustrate, suppose the historical cycle times are known at 81.6% and 89.8% utilizations. By re-running the simulations, the cycle times are 785.06 and 818.32 for 81.6% and 89.8%, respectively. Therefore, the values of k_2 can be calculated at 80% and 90% utilizations. They are

3901.6 and 4592.1, respectively. The other k_2 values can be extrapolated accordingly as shown in Table 7.4.

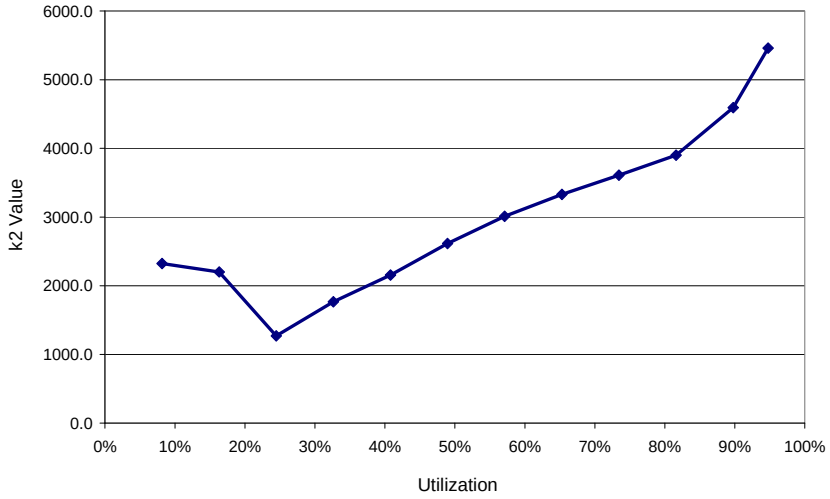

Figure 7.13 k2 values at different utilizations

Table 7.4 Cycle times from simulations and the single point k2 model

Input Rate (per day)	Utilization	Simulation				k2 Model	
		CT (min)	95% CI	k2	k2_Extrapolation	CT (min)	Error
20	8.2%	761.1	0.07	2325.8	-2437.3	761.0	-0.01%
40	16.3%	761.2	0.04	2202.2	-1733.0	761.0	-0.02%
60	24.5%	761.2	0.03	1269.8	-1028.6	760.9	-0.04%
80	32.6%	761.6	0.03	1765.9	-324.3	761.0	-0.07%
100	40.8%	762.2	0.03	2157.9	380.0	761.3	-0.11%
120	49.0%	763.3	0.03	2615.4	1084.3	762.1	-0.16%
140	57.1%	765.3	0.03	3011.7	1788.6	763.7	-0.21%
160	65.3%	768.4	0.03	3330.0	2493.0	766.7	-0.22%
180	73.4%	774.0	0.02	3610.7	3197.3	772.6	-0.18%
200	81.6%	785.1	0.04	3901.6	3901.6	785.1	0.00%
220	89.8%	818.2	0.12	4592.1	4605.9	818.3	0.02%
232	94.8%	892.8	0.43	5459.6	5037.6	883.2	-1.07%

The new approximate performance curve is shown in Figure 7.14, where the largest error is only -1.07% at 94.8% utilization. Since k_2 should not be negative, we can further improve the results by replacing the negative extrapolated values with zeros.

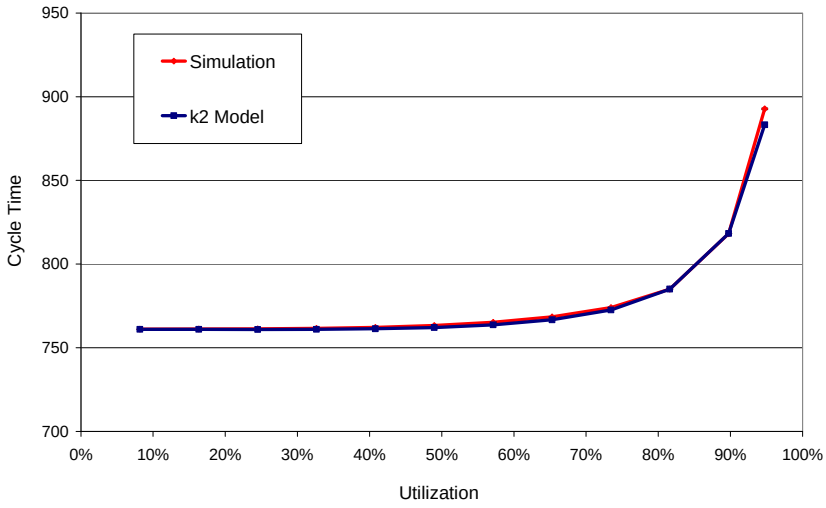

Figure 7.14 Performance curves based on two point k2 method

7.5 Conclusion

Comparing the three-parameter model with other approaches, Eq. (7.2) not only performs better but also permits some insight into those parameters. Because Eq. (7.2) and (7.7) describe the performance of a factory well, it can be used to quantify the performance of a factory, which is the goal we wanted to achieve at the very beginning of this thesis. With a long way from Chapter 2 to Chapter 7, we finally arrive at this destination.

In Section 7.3 and 7.4, we propose methods to approximate system performance by historical queueing times, without resorting to the second moment of the service time distribution. This important result constitutes a bridge between theory and practice.

Eq. (7.2) comes from capturing the essence, or underlying structures, of tandem queues. Although it performs well in the cases examined, it should be noted that the service time SCVs of these simulation models are small, because the service times are triangular distributed. When the service time SCV becomes larger, the one point intrinsic

ratio approximation may not suffice. We may resort to the two point approximation as we have demonstrated in Section 7.4.1.1.

In practice, the dispatching rules can be more complex than FCFS, earliest due date or the shortest remaining processing time. The planner may dispatch jobs "intelligently" by looking at the congestion levels of downstream workstations. The performance of Eq. (7.2) in this situation has not been verified.

In practice, the product mix can be complex and change everyday. Steady state may never be achieved. The performance of Eq. (7.2) in this situation is unknown.

The synchronization effects from batching and shift schedules are commonly seen in practical manufacturing systems. When the queueing time is dominated by the synchronization effects instead of the randomness effects, Eq. (7.2) has to be modified. For example, when the initial arrival batch exists, we can add an extra term, $(k_4 - 1)/2\lambda$, into Eq. (7.2), where k_4 is a parameter reflecting the batching effect.

In long term planning, knowing the overall system cycle time may be enough. However, in scheduling, we may be concerned about the cycle time performance of a specific product instead of the overall system cycle time. Different products may have different priorities at each workstation. In this case, further extensions and modifications on Eq. (7.2) are needed. This topic will be left as a direction of future research.

CHAPTER 8

CONCLUSION

The only real valuable thing is intuition. ~ Albert Einstein

Due to the complex situations that arise in practical manufacturing systems, understanding the behavior of even a single machine system can be complicated. In Chapter 2, based on a comprehensive classification of interruptions, proper queueing models are suggested for each type of interruption. In Chapter 3, a more accurate approximate model for general parallel batching process is proposed.

To understanding the behavior of a general queueing network, we begin by studying the underlying structure of simple tandem queues. The intrinsic gap and intrinsic ratio are observed. Based on the heavy traffic property of the intrinsic gap and the nearly-linear relationship of the intrinsic ratio, better approximate models are developed, which give smaller errors in the examined cases.

The correlations among workstations in general queueing networks are intricate. Previous approaches for analyzing general queuing networks require stringent assumptions which usually do not hold in practice. While generic models, such as Brownian motion, may be used to approximate the behavior of queuing systems, we also show that queuing systems have their own structure, the intrinsic ratio. The power of the intrinsic ratio is the approximately linear property. In this thesis, we use the intrinsic ratio to develop a new approach to the class of tandem queues, which deals with the correlations between workstations by using the underlying structure directly rather than making the stringent assumptions used in prior approaches. In Chapter 6, we further provide a way to analyze the correlations among general tandem queues through the viewpoint of the ASIA system.

In order to optimize the performance of a factory, we first need to quantify its performance. Based on the underlying structure of tandem queues, we developed an approximate model to describe the behavior of a factory by capturing the trade-off between the cycle time and utilization. The approximate model not only gives small errors in the examined cases, but also is more useful for practical applications, since it can predict future performance based on historical queueing time instead of requiring knowledge of the service time variability which may be very difficult to obtain for practical manufacturing systems.

One of the main motivations of this thesis was the desire to create methods for solving practical problems in manufacturing systems using theoretically sound approaches. If this desire could be achieved, it would help to eliminate the gap between theoretical approaches and practical experience. As we have shown, identifying the powerful underlying structure plays a key role. This thesis has demonstrated several basic problems in practical manufacturing systems, which have been analyzed by taking advantage of the underlying structure of the systems. There remain many similar challenges. It is our hope that this thesis will encourage more interest in exploring the useful underlying structure of manufacturing systems.

It is important for researchers to develop theories which may be useful in the future, but it is equally important to solve the problems which we encounter right now. Dantzig wrote in 1991, “it is interesting to note that the original problem that started my research is still outstanding – namely the problem of planning or scheduling dynamically over time, particularly planning dynamically under uncertainty. If such a problem could be successfully solved, it could eventually through better planning contribute to the well-being and stability of the world.” Solving practical problems is not easy but it is essential. Following Dantzig’s example, we hope this thesis contributes towards the goal of solving the original problem of predicting factory performance.

APPENDICES

APPENDIX A

A.1 Derivations of $E(G_1)$ for the Integrated Models

$$E(G_1)=\int_0^\infty E\left[G_1\middle|S_1=x\right]f_{s_1}(x)dx.$$

By Eq. (2.36),

$$E(G_1)=\int_0^\infty E\left[S_1+\sum_{i=1}^{N_r(S_1)}D_{r\_i}+\sum_{i=1}^{N_t(S_1)}D_{t\_i}+T_p\middle|S_1=x\right]f_{s_1}(x)dx$$

$$=\int_0^\infty x\,f_{s_1}(x)dx+\int_0^\infty E\left[\sum_{i=1}^{N_r(x)}D_{r\_i}\right]f_{s_1}(x)dx+\int_0^\infty E\left[\sum_{i=1}^{N_t(x)}D_{t\_i}\right]f_{s_1}(x)dx+\int_0^\infty E\left[T_p\right]f_{s_1}(x)dx$$

$$=E(S_1)+\int_0^\infty\sum_{n=0}^{\infty}E\left[\sum_{i=1}^{N_r(x)}D_{r\_i}\middle|N_r(x)=n\right]e^{-\eta_r x}\frac{(\eta_r x)^n}{n!}f_{s_1}(x)dx$$

$$+\int_0^\infty\sum_{n=0}^{\infty}E\left[\sum_{i=1}^{N_t(x)}D_{t\_i}\middle|N_t(x)=n\right]e^{-\eta_t x}\frac{(\eta_t x)^n}{n!}f_{s_1}(x)dx+E\left[T_p\right]$$

$$=E(S_1)+\int_0^\infty\sum_{n=0}^{\infty}E[n]E[D_r]\,e^{-\eta_r x}\frac{(\eta_r x)^n}{n!}f_{s_1}(x)dx+\int_0^\infty\sum_{n=0}^{\infty}E[n]E[D_t]e^{-\eta_t x}\frac{(\eta_t x)^n}{n!}f_{s_1}(x)dx+E\left[T_p\right]$$

$$=E(S_1)+\int_0^\infty\sum_{n=0}^{\infty}\eta_r xE[D_r]\,e^{-\eta_r x}\frac{(\eta_r x)^n}{n!}f_{s_1}(x)dx+\int_0^\infty\sum_{n=0}^{\infty}\eta_t xE[D_t]e^{-\eta_t x}\frac{(\eta_t x)^n}{n!}f_{s_1}(x)dx+E\left[T_p\right]$$

$$=E(S_1)+\int_0^\infty\eta_r xE[D_r]\,f_{s_1}(x)dx+\int_0^\infty\eta_t xE[D_t]\,f_{s_1}(x)dx+E\left[T_p\right]$$

$$=E(S_1)+E(S_1)\eta_r E(D_r)+E(S_1)\eta_t E(D_t)+E(T_p).$$

A.2 Derivations of $E(G_1^2)$ for the Integrated Models

$$E(G_1^2)=E\left[(S_1+\sum_{i=1}^{N_r(S_1)}D_{r\_i}+\sum_{i=1}^{N_t(S_1)}D_{t\_i}+T_p)^2\right]=\int_0^\infty E\left[(S_1+\sum_{i=1}^{N_r(S_1)}D_{r\_i}+\sum_{i=1}^{N_t(S_1)}D_{t\_i}+T_p)^2\middle|S_1=x\right]f_{s_1}(x)dx$$

$$=\int_0^\infty E\left[x^2\right]f_{s_1}(x)dx+\int_0^\infty E\left[(\sum_{i=1}^{N_r(x)}D_{r\_i})^2\right]f_{s_1}(x)dx+\int_0^\infty E\left[(\sum_{i=1}^{N_t(x)}D_{t\_i})^2\right]f_{s_1}(x)dx+\int_0^\infty E\left[T_p^2\right]f_{s_1}(x)dx$$

$$+\int_0^\infty E\left[2x\sum_{i=1}^{N_r(x)}D_{r\_i}\right]f_{s_1}(x)dx+\int_0^\infty E\left[2x\sum_{i=1}^{N_t(x)}D_{t\_i}\right]f_{s_1}(x)dx+\int_0^\infty E\left[2xT_p\right]f_{s_1}(x)dx$$

$$+\int_0^\infty E\left[2T_p\sum_{i=1}^{N_r(x)}D_{r\_i}\right]f_{s_1}(x)dx+\int_0^\infty E\left[2T_p\sum_{i=1}^{N_t(x)}D_{t\_i}\right]f_{s_1}(x)dx$$

$$+\int_0^\infty E\left[2(\sum_{i=1}^{N_r(x)}D_{r\_i})(\sum_{i=1}^{N_t(x)}D_{t\_i})\right]f_{s_1}(x)dx$$

$$= \int_0^\infty x^2 \, f_{s_1}(x)dx + \int_0^\infty E\left[(\sum_{i=1}^{N_r(x)} D_{r\_i})^2 \right] f_{s_1}(x)dx + \int_0^\infty E\left[(\sum_{i=1}^{N_t(x)} D_{t\_i})^2 \right] f_{s_1}(x)dx + \int_0^\infty E\left[T_p^2\right] f_{s_1}(x)dx$$

$$+2\int_0^\infty xE\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \right] f_{s_1}(x)dx + 2\int_0^\infty xE\left[\sum_{i=1}^{N_t(x)} D_{t\_i} \right] f_{s_1}(x)dx + 2E(T_p)\int_0^\infty x \, f_{s_1}(x)dx$$

$$+2E(T_p)\int_0^\infty E\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \right] f_{s_1}(x)dx + 2E(T_p)\int_0^\infty E\left[\sum_{i=1}^{N_t(x)} D_{t\_i} \right] f_{s_1}(x)dx$$

$$+2\int_0^\infty E\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \sum_{i=1}^{N_t(x)} D_{t\_i} \right] f_{s_1}(x)dx$$

$$= (1)+(2)+(3)+(4)+(5)+(6)+(7)+(8)+(9)+(10),$$

where

$$(1) = \int_0^\infty x^2 \, f_{s_1}(x)dx = E(S_1^2),$$

$$(2) = \int_0^\infty E\left[(\sum_{i=1}^{N_r(x)} D_{r\_i})^2 \right] f_{s_1}(x)dx = \int_0^\infty \sum_{n=0}^{\infty} E\left[(\sum_{i=1}^{N_r(x)} D_{r\_i})^2 \middle| N_r(x) = n \right] e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx$$

$$= \int_0^\infty \sum_{n=0}^{\infty} E\left[\sum_{i=1}^{n} D_{r\_i} \sum_{j=1}^{n} D_{r\_j} \right] e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx = \int_0^\infty \sum_{n=0}^{\infty} \sum_{i=1}^{n} \sum_{j=1}^{n} E\left[D_{r\_i} D_{r\_j} \right] e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx$$

$$= \int_0^\infty \sum_{n=0}^{\infty} \left[nE(D_r^2) + n(n-1)E(D_r)^2 \right] e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx$$

$$= \int_0^\infty E(D_r^2) \, e^{-\eta_r x} \sum_{n=0}^{\infty} n \, \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx \;\; + \int_0^\infty E(D_r)^2 e^{-\eta_r x} \sum_{n=0}^{\infty} n(n-1) \, \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx$$

$$= \int_0^\infty E(D_r^2) \, e^{-\eta_r x} \sum_{n=1}^{\infty} \frac{(\eta_r x)^{n-1}}{(n-1)!} \, \eta_r x \, f_{s_1}(x)dx + \int_0^\infty E(D_r)^2 e^{-\eta_r x} \sum_{n=2}^{\infty} \frac{(\eta_r x)^{n-2}}{(n-2)!} \, (\eta_r x)^2 \, f_{s_1}(x)dx$$

$$= \int_0^\infty E(D_r^2)\eta_r x f_{s_1}(x)dx + \int_0^\infty E(D_r)^2 (\eta_r x)^2 \, f_{s_1}(x)dx$$

$$= E(D_r^2)\eta_r E(S_1) + E(D_r)^2 \eta_r^2 E(S_1^2),$$

$$(3) = \int_0^\infty E\left[(\sum_{i=1}^{N_t(x)} D_{t\_i})^2 \right] f_{s_1}(x)dx$$

$$= E(D_t^2) \; \eta_t E(S_1) + E(D_t)^2 \eta_t^2 E(S_1^2), \text{ (same derivation as (2))}$$

$$(4) = \int_0^\infty E(T_p^2) \, f_{s_1}(x)dx = E(T_p^2),$$

$$(5) = 2\int_0^\infty xE\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \right] f_{s_1}(x)dx = 2\int_0^\infty x \sum_{n=0}^{\infty} E\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \middle| N_r(x) = n \right] e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx$$

$$= 2\int_0^\infty x \sum_{n=0}^{\infty} E[n]E[D_r] \, e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx = 2\int_0^\infty x \sum_{n=0}^{\infty} \eta_r x E[D_r] \, e^{-\eta_r x} \frac{(\eta_r x)^n}{n!} f_{s_1}(x)dx$$

$$= 2\int_0^\infty x^2 \eta_r E(D_r) f_{s_1}(x)dx = 2E(D_r)\eta_r E(S_1^2),$$

$$(6) = 2\int_0^\infty xE\left[\sum_{i=1}^{N_t(x)} D_{t\_i} \right] f_{s_1}(x)dx = 2E(D_t)\eta_t E(S_1^2),$$

$$(7) = 2E(T_p)\int_0^\infty x\, f_{s_1}(x)dx = 2E(T_p)E(S_1),$$

$$(8) = 2E(T_p)\int_0^\infty E\left[\sum_{i=1}^{N_r(x)} D_{r\_i}\right] f_{s_1}(x)dx = 2E(T_p)E(D_r)\eta_r E(S_1),$$

$$(9) = 2E(T_p)\int_0^\infty E\left[\sum_{i=1}^{N_t(x)} D_{t\_i}\right] f_{s_1}(x)dx = 2E(T_p)E(D_t)\eta_t E(S_1),$$

$$(10) = 2\int_0^\infty E\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \sum_{i=1}^{N_t(x)} D_{t\_i}\right] f_{s_1}(x)dx$$

$$= 2\int_0^\infty \sum_{n_r=0}^{\infty}\sum_{n_t=0}^{\infty} E\left[\sum_{i=1}^{N_r(x)} D_{r\_i} \sum_{i=1}^{N_t(x)} D_{t\_i} \,\middle|\, N_r(x)=n_r, N_t(x)=n_t\right] e^{-\eta_r x}\frac{(\eta_r x)^{n_r}}{n_r!} e^{-\eta_t x}\frac{(\eta_t x)^{n_t}}{n_t!} f_{s_1}(x)dx$$

$$= 2\int_0^\infty \sum_{n_r=0}^{\infty}\sum_{n_t=0}^{\infty} E(n_r)E(D_r)E(n_t)E(D_t)\, e^{-\eta_r x}\frac{(\eta_r x)^{n_r}}{n_r!} e^{-\eta_t x}\frac{(\eta_t x)^{n_t}}{n_t!} f_{s_1}(x)dx$$

$$= 2\int_0^\infty \sum_{n_r=0}^{\infty}\sum_{n_t=0}^{\infty} \eta_r x E(D_r)\eta_t x E(D_t)\, e^{-\eta_r x}\frac{(\eta_r x)^{n_r}}{n_r!} e^{-\eta_t x}\frac{(\eta_t x)^{n_t}}{n_t!} f_{s_1}(x)dx$$

$$= 2\int_0^\infty \eta_r x E(D_r)\eta_t x E(D_t) f_{s_1}(x)dx$$

$$= 2\eta_r E(D_r)\eta_t E(D_t)\int_0^\infty x^2 f_{s_1}(x)dx = 2\eta_r E(D_r)\eta_t E(D_t)E(S_1^2).$$

Therefore,

$$\begin{aligned}
&E(G_1^2) = (1)+(2)+(3)+(4)+(5)+(6)+(7)+(8)+(9)+(10)\\
&= E(S_1^2)+E(D_r^2)\eta_r E(S_1)+E(D_r)^2\eta_r^2 E(S_1^2)+E(D_t^2)\ \eta_t E(S_1)+E(D_t)^2\eta_t^2 E(S_1^2)\\
&+E(T_p^2)+2E(D_r)\eta_r E(S_1^2)+2E(D_t)\eta_t E(S_1^2)+2E(T_p)E(S_1)+2E(T_p)E(D_r)\eta_r E(S_1)\\
&+2E(T_p)E(D_t)\eta_t E(S_1)+2\eta_r E(D_r)\eta_t E(D_t)E(S_1^2)\\
&= E(S_1^2)\left\{\left[1+\eta_r \mathrm{E}(D_r)\right]^2+\left[1+\eta_t \mathrm{E}(D_t)\right]^2+2\mathrm{E}(T_p)-1\right\}\\
&+E(S_1)\eta_r\left[E(D_r^2)+2\mathrm{E}(T_p)\mathrm{E}(D_r)\right]+E(S_1)\eta_t\left[E(D_t^2)+2\mathrm{E}(T_p)\mathrm{E}(D_t)\right]+\mathrm{E}(T_p^2).
\end{aligned} \tag{2.52}$$

A.3 Derivations for M/G/1_Run-based Preemptive Event Model

$$E(QT) = \frac{\rho_G E(R_G)}{(1-\rho_G)} = \frac{\rho_G}{(1-\rho_G)}\frac{E(G^2)}{2E(G)},$$

where $E(G^2)$ can be got by assuming D_t and T_p to be zero in Eq. (2.52). Therefore,

$$E(QT)=\frac{\rho_G}{2(1-\rho_G)}\frac{E(S^2)\left[1+\eta E(D)\right]^2+\eta E(S)E(D^2)}{E(S)/A}=\frac{\rho_G}{2(1-\rho_G)}\frac{A}{E(S)}\left[\frac{E(S^2)}{A^2}+\eta E(S)E(D^2)\right]$$

$$=\frac{\rho_G}{2(1-\rho_G)}\frac{E(S)}{A}\frac{A}{E(S)}\left[\frac{E(S^2)}{E(S)A}+\eta AE(D^2)\right]=\frac{\rho_G}{2(1-\rho_G)}E(EPT)\frac{1}{E(S)}\left[\frac{E(S^2)}{E(S)}+\frac{\eta AE(D^2)}{1+\eta E(D)}\right]$$

$$=\frac{\rho_G}{2(1-\rho_G)}E(EPT)\left[\frac{E(S^2)}{E(S)^2}+\frac{A(1-A)E(D^2)}{E(S)E(D)}\right]$$

$$=\frac{\rho_G}{2(1-\rho_G)}E(EPT)\left[1+\frac{E(S^2)-E(S)^2}{E(S)^2}+\frac{A(1-A)E(D)}{E(S)}\left(1+\frac{E(D^2)-E(D)^2}{E(D)^2}\right)\right]$$

$$=\frac{\rho_G}{2(1-\rho_G)}E(EPT)\left[1+\frac{\sigma_0^2}{t_0^2}+\frac{A(1-A)m_r}{t_0}\left(1+\frac{\sigma_r^2}{m_r^2}\right)\right]$$

$$=\frac{\rho_G}{2(1-\rho_G)}E(EPT)\left[1+c_0^2+\frac{m_r}{t_0}\left(1+c_r^2\right)A(1-A)\right]$$

$$=\left(\frac{1+c_e^2}{2}\right)\left(\frac{\rho_G}{1-\rho_G}\right)E(EPT),$$

where A is availability, t_0 is mean service time, m_r is MTTR, σ_0 is the standard deviation of t_0 and σ_r is the standard deviation of m_r.

A.4 Derivations for M/G/1_ Run-based Non-Preemptive Product-Induced Setup Model

$$E(QT)=\frac{\rho_G E(R_G)}{(1-\rho_G)}=\frac{\rho_G}{(1-\rho_G)}\frac{E(G^2)}{2E(G)}=\frac{\rho_G}{(1-\rho_G)}\frac{E(S^2)+2E(S)E(T_p)+E(T_p^2)}{2E(G)}$$

$$=\frac{\rho_G}{2t_e(1-\rho_G)}\left(E(S^2)-E(S)^2+E(S)^2+2E(S)E(T_p)+E(T_p^2)\right)$$

$$=\frac{\rho_G}{2t_e(1-\rho_G)}\left(\sigma_0^2+t_0^2+2t_0\frac{t_p}{N_p}+E(T_p^2)\right),$$

where

$$E(T_p^2)=\frac{1}{N_p}\left(\sigma_p^2+t_p^2\right)+\left(1-\frac{1}{N_p}\right)0.$$

Therefore,

$$E(QT) = \frac{\rho_G}{2t_e(1-\rho_G)}\left(\sigma_0^2 + t_0^2 + 2t_0\frac{t_p}{N_p} + \frac{1}{N_p}\left(\sigma_p^2 + t_p^2\right)\right)$$

$$= \frac{\rho_G}{2t_e(1-\rho_G)}\left(\sigma_0^2 + t_0^2 + 2t_0\frac{t_p}{N_p} + \frac{t_p^2}{N_p^2} + \frac{\sigma_p^2}{N_p} + \frac{t_p^2}{N_p} - \frac{t_p^2}{N_p^2}\right)$$

$$= \frac{\rho_G}{2t_e(1-\rho_G)}\left(\left[t_0 + \frac{t_p}{N_p}\right]^2 + \sigma_0^2 + \frac{\sigma_p^2}{N_p} + \frac{t_p^2}{N_p} - \frac{t_p^2}{N_p^2}\right) = \frac{\rho_G}{2t_e(1-\rho_G)}\left(t_e^2 + \sigma_0^2 + \frac{\sigma_p^2}{N_p} + \frac{t_p^2}{N_p} - \frac{t_p^2}{N_p^2}\right)$$

$$= \frac{\rho_G}{2(1-\rho_G)}\left(1 + \frac{\sigma_0^2 + \frac{\sigma_p^2}{N_p} + \frac{t_p^2}{N_p} - \frac{t_p^2}{N_p^2}}{t_e^2}\right)t_e = \frac{1}{2}\left(1 + (\sigma_0^2 + \frac{\sigma_p^2}{N_p} + \frac{N_p - 1}{N_p^2}t_p^2)\right)\frac{\rho_G}{(1-\rho_G)}t_e$$

$$= \frac{(1+c_e^2)}{2}\frac{\rho_G}{(1-\rho_G)}E(EPT).$$

APPENDIX B

From Eq. (3.3), $\mu x^{k+1} - (\lambda + \mu)x + \lambda = 0,$

$$\mu(x - x^{k+1}) = \lambda(1 - x),$$
$$\mu x \frac{(1 - x^k)}{(1 - x)} = \lambda,$$
$$\frac{1}{k} \frac{x(1 - x^k)}{(1 - x)} = \frac{\lambda}{\mu k} = \rho,$$
$$\frac{1}{k} \sum_{n=1}^{k} x^n = \rho.$$

Let $f(x) = \frac{1}{k} \sum_{n=1}^{k} x^n$ and $g(\cdot) = f^{-1}(\cdot)$. Therefore, $x = g(\rho)$ and $f(x) = \rho$.

By using Taylor Series Expansion,

$$x = g(1) + (\rho - 1)g'(1) + \frac{1}{2}(\rho - 1)^2 g''(1) + O((1 - \rho)^3), \qquad \text{(B.1)}$$

where

$$\because f(g(x)) = x,\ f(g(1)) = 1 \text{ and } f(1) = 1,\ \Rightarrow g(1) = 1,$$

$$\because f'(g(x))g'(x) = 1,\ \therefore g'(1) = \frac{1}{f'(1)}$$

$$\because f'(x) = \frac{1}{k} \sum_{n=1}^{k} nx^n,\ \therefore f'(1) = \frac{1}{k} \sum_{n=1}^{k} n = \frac{1}{k} \frac{k(k+1)}{2} = \frac{k+1}{2},$$

$$\Rightarrow g'(1) = \frac{1}{k+1}.$$

$$\because f''(x) = \frac{1}{k} \sum_{n=1}^{k} n(n-1)x^n,$$

$$\therefore f'(1) = \frac{1}{k} \sum_{n=1}^{k} n(n-1) = \frac{1}{k} \frac{(k-1)k(k+1)}{3} = \frac{(k-1)(k+1)}{3},$$

$$\because f''(g(x))(g'(x))^2 +' f(g(x))g''(x) = 0,$$

$$\therefore f''(g(1))(g'(1))^2 + f'(g(1))g''(1) = 0,$$

$$\Rightarrow g''(1) = -\frac{f''(1)(g'(1))^2}{f'(1)} = -\frac{\frac{1}{3}(k-1)(k+1)(\frac{2}{k+1})^2}{\frac{k+1}{2}} = -\frac{8}{3}\frac{(k-1)}{(k+1)^2}.$$

Therefore, Eq. (B.1) becomes

$$\begin{aligned} x &= 1-\frac{2}{k+1}(1-\rho)-\frac{4}{3}\frac{(k-1)}{(k+1)^2}(1-\rho)^2+O((1-\rho)^3), \\ &\cong 1-\frac{2}{k+1}(1-\rho)-\frac{4}{3}\frac{(k-1)}{(k+1)^2}(1-\rho)^2. \end{aligned}$$

APPENDIX C

Results of STQB when service time SCV is smaller or equal to 1 (extreme cases):

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Exp-Const (1, 1, 0)				Exp-Const-Exp (1, 0, 1)				Exp-Const-Const (1, 0, 0)				Exp-Exp-Exp (1, 1, 1)			
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30
Intrinsic Ratio: (Sim QT – LB) / (UB – LB)	10%	100.0%	100.0%	100.0%	100.0%	18.8%	34.7%	41.5%	46.4%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	20%	100.0%	100.0%	100.0%	100.0%	17.1%	34.4%	41.8%	46.4%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	30%	100.0%	100.0%	100.0%	100.0%	19.8%	35.0%	41.8%	46.7%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	40%	100.0%	100.0%	100.0%	100.0%	18.6%	34.3%	41.0%	47.1%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	50%	100.0%	100.0%	100.0%	100.0%	21.1%	33.4%	40.4%	47.0%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	60%	100.0%	100.0%	100.0%	100.0%	15.4%	33.7%	41.3%	47.3%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	70%	100.0%	100.0%	100.0%	100.0%	19.1%	33.1%	39.7%	46.8%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	80%	100.0%	100.0%	100.0%	100.0%	25.5%	32.1%	39.5%	47.0%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	90%	100.0%	100.0%	100.0%	100.0%	29.4%	34.9%	33.7%	46.9%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
	95%	100.0%	100.0%	100.0%	100.0%	137.4%	31.8%	31.8%	40.5%	0.0%	0.0%	0.0%	0.0%	100.0%	100.0%	100.0%	100.0%
Sim QT of The 1st Server	10%	0.3	1.4	2.3	3.1	0.2	0.7	1.1	1.6	0.2	0.7	1.1	1.6	0.3	1.4	2.3	3.1
	20%	0.7	3.1	5.0	7.0	0.4	1.5	2.5	3.5	0.4	1.5	2.5	3.5	0.7	3.1	5.0	7.0
	30%	1.1	5.0	8.3	11.8	0.6	2.5	4.2	5.9	0.6	2.5	4.2	5.9	1.1	5.0	8.3	11.8
	40%	1.5	7.3	12.5	18.3	0.8	3.6	6.2	9.1	0.8	3.6	6.3	9.1	1.5	7.3	12.5	18.3
	50%	2.0	10.0	17.9	27.1	1.0	5.0	8.9	13.6	1.0	5.0	8.9	13.6	2.0	10.0	17.9	27.1
	60%	2.5	13.3	25.0	40.0	1.2	6.7	12.5	20.0	1.3	6.7	12.5	20.0	2.5	13.3	25.0	40.0
	70%	3.0	17.5	35.0	60.7	1.5	8.7	17.5	30.3	1.5	8.8	17.5	30.3	3.0	17.5	35.0	60.7
	80%	3.6	22.9	50.0	98.9	1.8	11.4	25.0	49.5	1.8	11.4	25.0	49.5	3.6	22.9	50.0	98.9
	90%	4.3	30.0	75.0	194.1	2.1	15.0	37.5	97.3	2.1	15.0	37.5	97.0	4.3	30.0	75.0	194.1
	95%	4.6	34.5	95.0	326.1	2.3	17.3	47.5	162.6	2.3	17.3	47.5	163.1	4.6	34.5	95.0	326.1
Sim QT of The 2nd Server	10%	1.7	1.7	1.7	1.7	3.2	2.9	2.7	2.5	1.5	1.0	0.5	0.1	3.3	3.3	3.3	3.3
	20%	3.8	3.8	3.8	3.8	7.2	6.5	6.0	5.6	3.4	2.2	1.3	0.3	7.5	7.5	7.5	7.5
	30%	6.4	6.4	6.4	6.4	12.4	11.2	10.4	9.7	5.9	3.9	2.3	0.5	12.9	12.9	12.9	12.9
	40%	10.0	10.0	10.0	10.0	19.4	17.6	16.3	15.2	9.2	6.4	3.8	0.9	20.0	20.0	20.0	20.0
	50%	15.0	15.0	15.0	15.0	29.2	26.7	24.7	22.8	14.0	10.0	6.1	1.4	30.0	30.0	30.0	30.0
	60%	22.5	22.5	22.5	22.5	43.9	40.6	37.7	34.4	21.3	15.8	10.0	2.5	45.0	45.0	45.0	45.0
	70%	35.0	35.0	35.0	35.0	68.8	64.1	59.4	53.9	33.5	26.3	17.5	4.7	70.0	70.0	70.0	70.0
	80%	60.0	60.0	60.0	60.0	118.6	112.2	104.9	93.8	58.2	48.6	35.0	10.5	120.0	120.0	120.0	120.0
	90%	135.0	135.0	135.0	135.0	268.5	260.2	245.1	218.5	132.9	120.0	97.5	38.0	270.0	270.0	270.0	270.0
	95%	285.0	285.0	285.0	285.0	570.9	558.2	537.6	473.0	282.7	267.7	237.5	121.9	570.0	570.0	570.0	570.0
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	82.9%	53.8%	42.3%	34.9%	94.9%	80.1%	70.1%	61.7%	89.7%	57.1%	31.8%	6.9%	90.6%	70.0%	59.5%	51.8%
	20%	84.0%	54.9%	42.9%	35.0%	95.3%	80.8%	70.7%	61.9%	90.5%	59.0%	33.3%	7.3%	91.3%	70.9%	60.0%	51.9%
	30%	85.3%	56.3%	43.5%	35.2%	95.7%	81.8%	71.5%	62.1%	91.4%	61.1%	35.2%	7.9%	92.0%	72.0%	60.7%	52.0%
	40%	86.7%	57.9%	44.4%	35.4%	96.2%	82.9%	72.3%	62.4%	92.3%	63.6%	37.5%	8.6%	92.9%	73.3%	61.5%	52.2%
	50%	88.2%	60.0%	45.7%	35.6%	96.7%	84.2%	73.5%	62.7%	93.3%	66.7%	40.5%	9.6%	93.8%	75.0%	62.7%	52.5%
	60%	90.0%	62.8%	47.4%	36.0%	97.2%	85.9%	75.1%	63.2%	94.4%	70.4%	44.4%	11.0%	94.7%	77.1%	64.3%	52.9%
	70%	92.0%	66.7%	50.0%	36.6%	97.8%	88.0%	77.2%	64.0%	95.7%	75.0%	50.0%	13.3%	95.8%	80.0%	66.7%	53.6%
	80%	94.3%	72.4%	54.5%	37.7%	98.5%	90.8%	80.7%	65.5%	97.0%	81.0%	58.3%	17.5%	97.1%	84.0%	70.6%	54.8%
	90%	96.9%	81.8%	64.3%	41.0%	99.2%	94.5%	86.7%	69.2%	98.4%	88.9%	72.2%	28.1%	98.4%	90.0%	78.3%	58.2%
	95%	98.4%	89.2%	75.0%	46.6%	99.6%	97.0%	91.9%	74.4%	99.2%	93.9%	83.3%	42.8%	99.2%	94.3%	85.7%	63.6%
Error % of 1st Approximate Models ($y = C_{s1}$)	10%	0.0%	0.0%	0.0%	0.0%	-1.0%	-8.7%	-17.7%	-28.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	20%	0.0%	0.0%	0.0%	0.0%	-0.8%	-8.2%	-17.3%	-28.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	30%	0.0%	0.0%	0.0%	0.0%	-0.9%	-7.8%	-16.7%	-28.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	40%	0.0%	0.0%	0.0%	0.0%	-0.7%	-7.1%	-15.7%	-28.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	50%	0.0%	0.0%	0.0%	0.0%	-0.7%	-6.3%	-14.6%	-28.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	60%	0.0%	0.0%	0.0%	0.0%	-0.4%	-5.5%	-13.7%	-27.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	70%	0.0%	0.0%	0.0%	0.0%	-0.4%	-4.5%	-11.7%	-26.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	80%	0.0%	0.0%	0.0%	0.0%	-0.4%	-3.3%	-9.4%	-24.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	90%	0.0%	0.0%	0.0%	0.0%	-0.2%	-2.0%	-5.2%	-20.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	95%	0.0%	0.0%	0.0%	0.0%	-0.6%	-1.0%	-2.8%	-14.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Error % of 2nd Approximate Models (y at $\rho = 80\%$)	10%	0.0%	0.0%	0.0%	0.0%	-8.3%	-20.9%	-20.5%	-11.7%	-15.5%	-37.0%	-14.3%	370.6%	0.0%	0.0%	0.0%	0.0%
	20%	0.0%	0.0%	0.0%	0.0%	-7.5%	-19.8%	-20.1%	-11.7%	-14.2%	-34.3%	-13.3%	347.4%	0.0%	0.0%	0.0%	0.0%
	30%	0.0%	0.0%	0.0%	0.0%	-6.9%	-18.8%	-19.3%	-11.8%	-12.7%	-31.4%	-12.3%	321.5%	0.0%	0.0%	0.0%	0.0%
	40%	0.0%	0.0%	0.0%	0.0%	-6.1%	-17.3%	-18.3%	-11.8%	-11.2%	-28.2%	-11.1%	292.4%	0.0%	0.0%	0.0%	0.0%
	50%	0.0%	0.0%	0.0%	0.0%	-5.3%	-15.5%	-17.0%	-11.6%	-9.6%	-24.7%	-9.8%	259.5%	0.0%	0.0%	0.0%	0.0%
	60%	0.0%	0.0%	0.0%	0.0%	-4.3%	-13.6%	-15.9%	-11.5%	-7.9%	-20.8%	-8.3%	222.1%	0.0%	0.0%	0.0%	0.0%
	70%	0.0%	0.0%	0.0%	0.0%	-3.4%	-11.2%	-13.6%	-10.9%	-6.1%	-16.4%	-6.7%	179.1%	0.0%	0.0%	0.0%	0.0%
	80%	0.0%	0.0%	0.0%	0.0%	-2.5%	-8.3%	-11.0%	-10.3%	-4.2%	-11.6%	-4.8%	129.0%	0.0%	0.0%	0.0%	0.0%
	90%	0.0%	0.0%	0.0%	0.0%	-1.3%	-4.9%	-6.2%	-8.6%	-2.2%	-6.2%	-2.6%	70.2%	0.0%	0.0%	0.0%	0.0%
	95%	0.0%	0.0%	0.0%	0.0%	-1.1%	-2.5%	-3.4%	-4.5%	-1.1%	-3.2%	-1.3%	36.7%	0.0%	0.0%	0.0%	0.0%
Error % of QNA	10%	0.0%	0.0%	0.0%	0.0%	4.3%	16.0%	24.5%	32.6%	11.4%	74.2%	212.1%	1335.7%	0.0%	0.0%	0.0%	0.0%
	20%	0.0%	0.0%	0.0%	0.0%	3.9%	14.5%	22.3%	30.6%	10.0%	66.6%	191.7%	1213.7%	0.0%	0.0%	0.0%	0.0%
	30%	0.0%	0.0%	0.0%	0.0%	3.1%	12.2%	19.4%	26.9%	8.4%	57.1%	166.4%	1063.5%	0.0%	0.0%	0.0%	0.0%
	40%	0.0%	0.0%	0.0%	0.0%	2.3%	9.5%	15.8%	22.0%	6.4%	46.0%	137.0%	890.4%	0.0%	0.0%	0.0%	0.0%
	50%	0.0%	0.0%	0.0%	0.0%	1.3%	6.2%	11.0%	16.1%	4.2%	33.3%	104.2%	700.8%	0.0%	0.0%	0.0%	0.0%
	60%	0.0%	0.0%	0.0%	0.0%	0.4%	2.0%	4.5%	8.7%	1.6%	19.4%	68.8%	503.0%	0.0%	0.0%	0.0%	0.0%
	70%	0.0%	0.0%	0.0%	0.0%	-1.0%	-2.8%	-2.3%	0.2%	-1.1%	4.3%	31.9%	307.6%	0.0%	0.0%	0.0%	0.0%
	80%	0.0%	0.0%	0.0%	0.0%	-2.5%	-8.3%	-11.0%	-10.3%	-4.2%	-11.6%	-4.8%	129.0%	0.0%	0.0%	0.0%	0.0%
	90%	0.0%	0.0%	0.0%	0.0%	-4.0%	-14.9%	-20.8%	-23.2%	-7.5%	-28.0%	-39.4%	-13.5%	0.0%	0.0%	0.0%	0.0%
	95%	0.0%	0.0%	0.0%	0.0%	-5.2%	-18.4%	-27.2%	-30.3%	-9.3%	-36.2%	-55.2%	-63.4%	0.0%	0.0%	0.0%	0.0%
Error % of QNET	10%	0.0%	0.0%	0.0%	0.0%	-21.4%	-10.4%	-3.6%	1.4%	-100.0%	-100.0%	-100.0%	-99.3%	0.0%	0.0%	0.0%	0.0%
	20%	0.0%	0.0%	0.0%	0.0%	-21.2%	-11.2%	-4.1%	1.4%	-100.0%	-100.0%	-100.0%	-99.7%	0.0%	0.0%	0.0%	0.0%
	30%	0.0%	0.0%	0.0%	0.0%	-20.9%	-12.4%	-4.6%	1.1%	-100.0%	-100.0%	-100.0%	-99.9%	0.0%	0.0%	0.0%	0.0%
	40%	0.0%	0.0%	0.0%	0.0%	-20.1%	-13.6%	-5.1%	0.8%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%
	50%	0.0%	0.0%	0.0%	0.0%	-19.1%	-15.0%	-5.9%	0.7%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%
	60%	0.0%	0.0%	0.0%	0.0%	-17.3%	-16.8%	-7.7%	0.4%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%
	70%	0.0%	0.0%	0.0%	0.0%	-15.1%	-18.4%	-9.7%	0.3%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%
	80%	0.0%	0.0%	0.0%	0.0%	-11.9%	-19.0%	-13.5%	-0.3%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%
	90%	0.0%	0.0%	0.0%	0.0%	-7.3%	-16.3%	-17.4%	-1.9%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%
	95%	0.0%	0.0%	0.0%	0.0%	-4.5%	-11.1%	-16.5%	-3.8%	-100.0%	-100.0%	-100.0%	-100.0%	0.0%	0.0%	0.0%	0.0%

APPENDIX D

Results of STQB when service time SCV is smaller than 1:

$(C_{a1}^2, C_{s1}^2, C_{s2}^2)$		Exp-Gam-Gam (1, 0.1, 0.1)				Exp-Gam-Gam (1, 0.1, 0.5)				Exp-Gam-Gam (1, 0.5, 0.1)				Exp-Gam-Gam (1, 0.5, 0.5)			
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30
Intrinsic Ratio: (Sim QT – LB) / (UB – LB)	10%	18.2%	24.7%	32.0%	39.0%	26.1%	38.3%	45.1%	50.4%	66.7%	73.5%	77.2%	79.7%	70.9%	77.2%	80.3%	82.5%
	20%	16.8%	23.6%	31.2%	38.6%	24.9%	37.9%	44.9%	50.0%	66.1%	72.2%	76.2%	79.1%	70.1%	76.8%	79.6%	81.5%
	30%	16.7%	22.6%	30.2%	38.0%	26.0%	37.5%	44.1%	49.9%	64.6%	71.3%	75.2%	78.4%	69.5%	75.5%	78.3%	81.0%
	40%	16.0%	21.1%	29.0%	37.5%	24.2%	35.8%	43.1%	49.4%	63.5%	69.6%	74.1%	77.8%	68.3%	74.4%	77.8%	80.5%
	50%	16.0%	20.2%	27.5%	36.9%	24.8%	35.1%	42.6%	49.3%	62.7%	67.8%	72.9%	77.1%	67.4%	73.3%	76.9%	80.0%
	60%	16.6%	18.4%	26.1%	36.2%	22.6%	33.9%	41.6%	48.9%	61.4%	65.9%	71.1%	76.2%	65.8%	71.7%	75.7%	79.2%
	70%	14.2%	17.5%	24.0%	35.2%	21.4%	31.9%	39.9%	48.5%	60.0%	63.5%	69.4%	75.4%	63.7%	70.0%	74.5%	78.6%
	80%	14.0%	16.2%	21.9%	33.8%	20.4%	30.0%	37.8%	47.2%	58.0%	59.6%	66.9%	74.3%	61.2%	67.1%	72.6%	77.4%
	90%	21.9%	12.1%	18.2%	30.5%	5.0%	30.4%	34.8%	44.3%	53.1%	58.7%	62.6%	72.3%	59.0%	65.7%	69.8%	76.7%
	95%	13.2%	6.2%	17.6%	26.2%	-18.0%	32.1%	36.4%	42.2%	97.8%	53.4%	54.4%	70.0%	56.6%	57.7%	60.7%	74.4%
Sim QT of The 1st Server	10%	0.2	0.8	1.2	1.7	0.2	0.8	1.2	1.7	0.3	1.1	1.7	2.3	0.3	1.1	1.7	2.3
	20%	0.4	1.7	2.8	3.8	0.4	1.7	2.8	3.8	0.5	2.3	3.7	5.2	0.5	2.3	3.8	5.2
	30%	0.6	2.8	4.6	6.5	0.6	2.7	4.6	6.5	0.8	3.8	6.2	8.9	0.8	3.7	6.2	8.9
	40%	0.8	4.0	6.9	10.1	0.8	4.0	6.9	10.1	1.2	5.5	9.4	13.7	1.2	5.5	9.4	13.7
	50%	1.1	5.5	9.8	14.9	1.1	5.5	9.8	14.9	1.5	7.5	13.4	20.4	1.5	7.5	13.4	20.4
	60%	1.4	7.3	13.8	22.0	1.4	7.3	13.8	22.0	1.9	10.0	18.8	30.0	1.9	10.0	18.8	30.1
	70%	1.7	9.6	19.2	33.4	1.7	9.6	19.3	33.4	2.3	13.1	26.2	45.5	2.3	13.1	26.3	45.5
	80%	2.0	12.6	27.5	54.5	2.0	12.6	27.5	54.5	2.7	17.1	37.5	74.3	2.7	17.1	37.5	74.0
	90%	2.4	16.5	41.3	106.7	2.4	16.5	41.2	106.5	3.2	22.5	56.2	145.9	3.2	22.5	56.2	145.6
	95%	2.5	19.0	52.2	178.8	2.5	19.0	52.3	179.2	3.5	25.9	71.1	245.0	3.5	25.9	71.4	245.4
Sim QT of The 2nd Server	10%	1.7	1.2	1.0	0.8	2.4	2.0	1.8	1.7	1.7	1.5	1.4	1.4	2.4	2.3	2.2	2.1
	20%	3.8	2.8	2.2	1.8	5.3	4.6	4.1	3.7	3.9	3.5	3.2	3.0	5.5	5.1	4.9	4.7
	30%	6.6	4.9	3.9	3.0	9.2	7.9	7.1	6.4	6.8	6.0	5.5	5.2	9.4	8.7	8.3	8.0
	40%	10.3	7.8	6.1	4.7	14.4	12.4	11.1	9.9	10.6	9.3	8.6	8.0	14.6	13.6	12.9	12.3
	50%	15.6	12.1	9.4	7.1	21.7	18.9	16.9	14.9	15.9	14.1	12.9	11.8	22.0	20.5	19.4	18.4
	60%	23.6	18.8	14.6	10.7	32.7	28.9	25.7	22.5	24.0	21.3	19.3	17.6	33.1	30.9	29.2	27.5
	70%	37.1	30.6	23.9	16.9	51.2	45.9	40.9	35.3	37.6	33.7	30.5	27.3	51.7	48.6	45.8	42.8
	80%	64.3	55.5	44.5	30.0	88.4	81.2	72.9	61.3	64.9	59.1	53.6	47.0	88.9	84.4	79.7	73.2
	90%	146.7	134.0	114.7	74.3	200.3	191.0	175.6	143.0	147.0	139.2	127.4	108.2	201.2	194.8	185.5	168.6
	95%	311.3	295.7	270.4	181.1	424.5	414.6	394.3	323.8	313.4	301.4	281.0	240.0	426.0	416.5	399.5	364.9
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	89.9%	61.3%	44.1%	31.7%	92.5%	72.0%	59.2%	49.1%	87.1%	59.1%	45.8%	36.9%	90.4%	67.8%	55.9%	47.3%
	20%	90.6%	62.6%	44.8%	31.7%	93.1%	73.0%	59.9%	49.2%	88.0%	60.1%	46.3%	36.8%	91.1%	68.8%	56.4%	47.2%
	30%	91.5%	64.2%	45.8%	31.8%	93.8%	74.2%	60.7%	49.5%	89.1%	61.5%	46.9%	36.8%	91.8%	69.9%	57.1%	47.3%
	40%	92.4%	66.2%	47.1%	31.9%	94.4%	75.7%	61.7%	49.6%	90.2%	63.1%	47.7%	36.7%	92.7%	71.4%	57.9%	47.4%
	50%	93.4%	68.7%	48.9%	32.2%	95.2%	77.5%	63.2%	50.0%	91.4%	65.3%	49.0%	36.8%	93.6%	73.2%	59.2%	47.5%
	60%	94.5%	71.9%	51.5%	32.7%	96.0%	79.8%	65.1%	50.5%	92.8%	68.1%	50.8%	37.0%	94.6%	75.6%	60.9%	47.8%
	70%	95.7%	76.1%	55.4%	33.6%	96.8%	82.7%	68.0%	51.4%	94.3%	72.0%	53.7%	37.5%	95.8%	78.7%	63.6%	48.4%
	80%	97.0%	81.5%	61.8%	35.5%	97.8%	86.6%	72.6%	52.9%	96.0%	77.5%	58.8%	38.7%	97.0%	83.1%	68.0%	49.7%
	90%	98.4%	89.0%	73.5%	41.0%	98.8%	92.0%	81.0%	57.3%	97.9%	86.1%	69.4%	42.6%	98.4%	89.6%	76.7%	53.7%
	95%	99.2%	94.0%	83.8%	50.3%	99.4%	95.6%	88.3%	64.4%	98.9%	92.1%	79.8%	49.5%	99.2%	94.2%	84.8%	59.8%
Error % of 1st Approximate Models ($y = C_s$)	10%	1.5%	4.4%	-0.5%	-15.8%	0.4%	-2.6%	-9.3%	-19.4%	0.6%	-1.9%	-7.7%	-15.3%	0.0%	-3.1%	-7.6%	-13.1%
	20%	1.5%	4.8%	0.6%	-15.0%	0.5%	-2.3%	-8.9%	-18.9%	0.6%	-1.0%	-6.4%	-14.4%	0.1%	-2.8%	-6.8%	-12.1%
	30%	1.4%	5.0%	1.7%	-13.7%	0.4%	-2.0%	-8.1%	-18.7%	0.8%	-0.4%	-5.1%	-13.3%	0.1%	-2.1%	-5.8%	-11.5%
	40%	1.3%	5.4%	3.0%	-12.6%	0.4%	-1.3%	-7.1%	-18.0%	0.8%	0.7%	-3.7%	-12.2%	0.2%	-1.5%	-5.2%	-10.9%
	50%	1.1%	5.2%	4.3%	-11.2%	0.3%	-1.0%	-6.4%	-17.7%	0.8%	1.5%	-2.2%	-11.0%	0.2%	-0.9%	-4.3%	-10.3%
	60%	0.9%	5.2%	5.3%	-9.4%	0.4%	-0.6%	-5.3%	-16.9%	0.7%	2.2%	-0.4%	-9.3%	0.3%	-0.3%	-3.2%	-9.3%
	70%	0.8%	4.5%	6.1%	-7.2%	0.3%	-0.1%	-3.9%	-15.9%	0.7%	2.8%	1.1%	-7.9%	0.3%	0.2%	-2.2%	-8.4%
	80%	0.5%	3.5%	6.0%	-4.0%	0.3%	0.3%	-2.3%	-13.8%	0.5%	3.2%	2.7%	-5.7%	0.3%	0.7%	-0.9%	-6.8%
	90%	0.2%	2.4%	4.8%	1.7%	0.3%	0.1%	-0.8%	-9.4%	0.4%	1.9%	3.6%	-2.2%	0.2%	0.6%	0.3%	-5.2%
	95%	0.2%	1.6%	2.7%	5.4%	0.3%	0.0%	-0.6%	-5.8%	-0.3%	1.5%	4.1%	0.8%	0.1%	0.8%	1.8%	-2.5%
Error % of 2nd Approximate Models (y at $\rho = 80\%$)	10%	-12.4%	-29.7%	-24.5%	3.7%	-9.5%	-23.6%	-22.3%	-10.1%	-6.7%	-16.1%	-15.1%	-6.6%	-5.2%	-12.8%	-12.5%	-7.4%
	20%	-11.3%	-27.4%	-22.7%	4.5%	-8.6%	-22.2%	-21.5%	-9.6%	-6.0%	-14.5%	-13.6%	-5.6%	-4.7%	-12.1%	-11.7%	-6.4%
	30%	-10.1%	-24.9%	-20.7%	5.6%	-7.8%	-20.7%	-20.3%	-9.4%	-5.3%	-13.2%	-12.2%	-4.5%	-4.2%	-10.9%	-10.5%	-5.7%
	40%	-8.9%	-22.1%	-18.3%	6.7%	-6.8%	-18.7%	-18.8%	-8.8%	-4.6%	-11.3%	-10.6%	-3.4%	-3.7%	-9.7%	-9.7%	-5.2%
	50%	-7.6%	-19.3%	-15.5%	7.8%	-5.9%	-16.6%	-17.4%	-8.6%	-3.9%	-9.4%	-8.8%	-2.2%	-3.1%	-8.4%	-8.6%	-4.6%
	60%	-6.3%	-15.8%	-12.6%	9.2%	-4.8%	-14.2%	-15.4%	-8.1%	-3.1%	-7.4%	-6.5%	-0.6%	-2.5%	-6.9%	-7.2%	-3.7%
	70%	-4.8%	-12.5%	-9.1%	10.7%	-3.7%	-11.3%	-12.8%	-7.4%	-2.3%	-5.2%	-4.3%	0.7%	-1.9%	-5.4%	-5.7%	-3.0%
	80%	-3.3%	-8.7%	-5.7%	12.4%	-2.5%	-8.1%	-9.4%	-5.8%	-1.5%	-2.7%	-1.7%	2.4%	-1.2%	-3.4%	-3.8%	-1.6%
	90%	-1.8%	-4.2%	-2.0%	14.7%	-1.1%	-4.5%	-5.2%	-2.7%	-0.7%	-1.4%	0.8%	4.7%	-0.6%	-1.8%	-1.6%	-0.8%
	95%	-0.9%	-1.8%	-0.9%	14.4%	-0.4%	-2.5%	-3.1%	-0.8%	-0.8%	-0.3%	2.6%	6.0%	-0.3%	-0.5%	0.7%	0.9%
Error % of QNA	10%	9.1%	47.1%	85.4%	129.9%	5.9%	23.7%	37.3%	50.4%	4.9%	18.1%	26.5%	34.2%	3.1%	10.7%	15.2%	19.1%
	20%	8.2%	43.5%	80.6%	125.0%	5.3%	21.7%	34.6%	48.0%	4.4%	17.5%	26.0%	33.6%	2.8%	9.8%	14.7%	19.2%
	30%	6.9%	38.4%	73.3%	117.0%	4.3%	18.8%	31.0%	43.6%	3.9%	15.8%	24.4%	31.9%	2.4%	9.0%	13.9%	17.9%
	40%	5.4%	32.1%	63.5%	104.6%	3.4%	15.5%	26.3%	37.8%	3.1%	14.0%	21.8%	28.8%	1.9%	7.6%	11.8%	15.6%
	50%	3.5%	23.8%	50.9%	88.2%	2.1%	10.9%	19.5%	29.5%	2.2%	11.2%	18.1%	24.5%	1.3%	5.7%	9.2%	12.6%
	60%	1.4%	14.6%	35.0%	67.8%	0.8%	5.6%	11.6%	19.8%	1.1%	7.5%	13.4%	19.1%	0.6%	3.3%	6.0%	8.9%
	70%	-0.8%	3.5%	16.4%	42.6%	-0.8%	-0.7%	2.1%	7.8%	-0.1%	2.9%	6.8%	11.6%	-0.2%	0.3%	1.6%	4.0%
	80%	-3.3%	-8.7%	-5.7%	12.4%	-2.5%	-8.1%	-9.4%	-5.8%	-1.5%	-2.7%	-1.7%	2.4%	-1.2%	-3.4%	-3.8%	-1.6%
	90%	-6.2%	-21.8%	-30.1%	-23.9%	-4.3%	-16.9%	-23.6%	-22.7%	-3.1%	-10.8%	-13.3%	-10.0%	-2.4%	-8.5%	-11.3%	-10.2%
	95%	-7.6%	-28.8%	-43.5%	-46.3%	-5.4%	-21.7%	-32.3%	-34.8%	-4.5%	-15.0%	-20.2%	-19.5%	-3.0%	-11.1%	-15.3%	-15.8%
Error % of QNET	10%	-76.5%	-48.6%	-26.2%	-19.6%	-15.4%	-25.5%	-11.0%	-4.0%	-25.7%	-12.7%	-9.7%	-7.1%	-10.5%	-6.6%	-5.2%	-4.1%
	20%	-77.2%	-51.7%	-26.4%	-19.3%	-55.2%	-27.8%	-11.7%	-3.8%	-26.5%	-12.2%	-8.8%	-6.2%	-10.3%	-6.5%	-4.7%	-3.0%
	30%	-77.9%	-55.3%	-27.2%	-18.7%	-56.3%	-30.6%	-12.5%	-3.8%	-27.2%	-12.3%	-8.1%	-5.2%	-10.1%	-6.2%	-3.9%	-2.5%
	40%	-78.8%	-59.2%	-29.0%	-18.3%	-57.2%	-33.7%	-13.7%	-3.5%	-27.8%	-12.3%	-7.3%	-4.4%	-9.8%	-6.0%	-3.7%	-2.1%
	50%	-80.0%	-63.4%	-32.5%	-17.9%	-57.7%	-37.7%	-16.1%	-3.8%	-28.0%	-13.0%	-6.6%	-3.4%	-9.2%	-6.0%	-3.3%	-1.6%
	60%	-82.0%	-67.3%	-38.6%	-17.2%	-57.4%	-42.3%	-19.8%	-3.9%	-27.4%	-14.6%	-5.9%	-2.2%	-8.4%	-6.1%	-2.9%	-0.9%
	70%	-85.1%	-71.2%	-48.0%	-16.3%	-55.7%	-47.4%	-25.7%	-4.4%	-25.6%	-17.4%	-6.2%	-1.5%	-7.3%	-6.6%	-2.8%	-0.5%
	80%	-89.4%	-75.0%	-60.1%	94.7%	-50.9%	-53.0%	-35.3%	-5.2%	-21.7%	-21.5%	-8.5%	-0.8%	-5.7%	-7.1%	-3.3%	0.4%
	90%	-94.2%	-81.0%	-71.6%	-21.4%	-39.1%	-56.3%	-49.0%	-9.8%	-14.7%	-24.9%	-16.9%	-0.7%	-3.5%	-6.6%	-5.4%	-0.3%
	95%	-96.4%	-89.1%	-78.1%	-45.0%	-26.4%	-50.3%	-55.8%	-24.1%	-9.6%	-20.0%	-22.4%	-3.0%	-2.0%	-4.3%	-5.0%	-0.6%

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 0.9, 0.5)				Exp-Gam-Gam (1, 0.5, 0.9)				Exp-Gam-Gam (1, 0.9, 0.9)				Exp-Gam-Gam (1, 0.1, 0.9)			
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30
Intrinsic Ratio: (Sim QT – LB) / (UB – LB)	10%	97.4%	96.6%	97.0%	97.5%	73.2%	80.3%	82.5%	83.8%	95.8%	96.7%	97.5%	97.6%	34.2%	46.6%	51.8%	56.2%
	20%	94.6%	96.4%	96.9%	97.3%	72.1%	79.0%	81.8%	83.6%	95.8%	96.7%	97.3%	97.3%	33.7%	45.9%	51.8%	56.4%
	30%	95.0%	96.1%	96.8%	97.1%	72.7%	78.4%	81.2%	83.0%	95.6%	96.8%	97.0%	97.3%	32.9%	44.6%	51.7%	56.4%
	40%	94.3%	96.3%	96.6%	96.9%	73.1%	77.8%	80.0%	82.6%	95.2%	96.0%	96.7%	97.2%	30.4%	43.6%	50.5%	55.8%
	50%	94.6%	95.7%	96.4%	96.9%	71.5%	76.4%	79.3%	81.8%	96.7%	95.7%	97.0%	97.1%	30.4%	43.5%	50.0%	55.4%
	60%	95.3%	95.6%	96.9%	96.8%	70.1%	74.8%	78.3%	81.1%	95.0%	95.4%	96.5%	97.3%	27.3%	43.0%	49.2%	55.2%
	70%	95.2%	94.8%	95.5%	96.5%	70.1%	73.8%	76.7%	80.7%	94.7%	96.1%	96.4%	97.0%	28.2%	40.0%	48.6%	55.1%
	80%	92.1%	94.6%	95.5%	96.9%	64.4%	73.3%	75.7%	79.9%	90.5%	95.3%	95.1%	96.2%	19.4%	40.7%	46.3%	54.0%
	90%	96.3%	93.0%	94.3%	96.2%	77.1%	72.1%	74.0%	78.4%	86.4%	94.1%	96.3%	96.3%	16.2%	39.0%	42.3%	52.6%
	95%	106.3%	94.4%	94.7%	96.1%	46.0%	73.8%	64.4%	78.3%	77.1%	95.1%	96.0%	95.6%	11.6%	32.2%	36.7%	50.4%
Sim QT of The 1st Server	10%	0.3	1.4	2.2	3.0	0.3	1.1	1.7	2.3	0.3	1.4	2.2	2.9	0.2	0.8	1.2	1.7
	20%	0.7	2.9	4.7	6.6	0.5	2.3	3.7	5.2	0.7	2.9	4.8	6.6	0.4	1.7	2.7	3.8
	30%	1.1	4.7	7.9	11.3	0.8	3.8	6.3	8.9	1.1	4.8	7.9	11.3	0.6	2.7	4.6	6.5
	40%	1.5	6.9	11.9	17.3	1.2	5.5	9.4	13.7	1.5	6.9	11.9	17.3	0.8	4.0	6.9	10.1
	50%	1.9	9.5	17.0	25.8	1.5	7.5	13.4	20.3	1.9	9.5	17.0	25.8	1.1	5.5	9.8	14.9
	60%	2.4	12.7	23.8	38.0	1.9	10.0	18.7	30.0	2.4	12.7	23.8	38.0	1.4	7.3	13.7	22.0
	70%	2.9	16.6	33.2	57.7	2.3	13.1	26.2	45.4	2.9	16.6	33.2	57.6	1.7	9.6	19.3	33.4
	80%	3.5	21.7	47.5	94.3	2.7	17.2	37.5	74.2	3.5	21.8	47.4	94.0	2.0	12.6	27.4	54.5
	90%	4.1	28.5	71.5	185.0	3.2	22.5	56.2	146.0	4.1	28.5	71.3	185.0	2.4	16.5	41.2	106.9
	95%	4.4	32.8	90.3	310.9	3.5	25.9	71.2	243.4	4.4	32.8	90.4	311.3	2.5	19.0	52.2	178.6
Sim QT of The 2nd Server	10%	2.5	2.5	2.4	2.4	3.1	3.0	2.9	2.8	3.2	3.1	3.1	3.1	3.0	2.7	2.6	2.4
	20%	5.6	5.5	5.5	5.4	7.0	6.6	6.4	6.3	7.1	7.0	7.0	6.9	6.9	6.2	5.8	5.5
	30%	9.6	9.5	9.4	9.3	12.0	11.4	11.0	10.7	12.2	12.1	12.0	11.9	11.8	10.7	10.0	9.4
	40%	14.9	14.7	14.6	14.5	18.7	17.8	17.1	16.6	18.9	18.7	18.6	18.5	18.4	16.7	15.6	14.6
	50%	22.4	22.1	21.9	21.7	28.1	26.7	25.7	24.8	28.4	28.1	28.0	27.8	27.7	25.4	23.6	21.8
	60%	33.6	33.2	33.0	32.5	42.2	40.2	38.7	37.1	42.6	42.2	41.9	41.7	41.8	38.6	35.8	32.9
	70%	52.4	51.6	51.0	50.5	65.8	63.1	60.4	57.7	66.3	65.8	65.3	64.7	65.3	60.7	56.6	51.5
	80%	89.7	88.8	87.9	87.1	113.0	109.4	104.9	99.1	113.7	113.0	111.7	110.4	112.4	106.5	99.2	89.0
	90%	202.3	200.5	198.5	195.5	255.8	250.2	241.9	225.1	255.9	254.8	253.8	249.7	254.5	246.4	232.7	205.9
	95%	427.8	425.7	422.7	415.4	539.6	534.7	516.1	488.3	540.5	539.9	537.9	527.9	539.2	528.6	508.4	452.6
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	88.4%	64.4%	53.1%	45.1%	92.3%	73.4%	62.8%	54.5%	90.6%	69.7%	59.0%	51.2%	94.1%	77.7%	67.2%	58.6%
	20%	89.2%	65.4%	53.6%	45.2%	92.9%	74.2%	63.2%	54.6%	91.3%	70.6%	59.5%	51.3%	94.6%	78.6%	67.8%	58.7%
	30%	90.1%	66.6%	54.3%	45.3%	93.5%	75.2%	63.8%	54.6%	92.0%	71.7%	60.2%	51.4%	95.1%	79.5%	68.5%	59.0%
	40%	91.1%	68.1%	55.1%	45.5%	94.2%	76.5%	64.6%	54.8%	92.8%	73.0%	61.1%	51.7%	95.6%	80.7%	69.4%	59.1%
	50%	92.2%	69.9%	56.4%	45.7%	94.9%	78.1%	65.8%	54.9%	93.7%	74.8%	62.3%	51.9%	96.2%	82.2%	70.6%	59.4%
	60%	93.4%	72.4%	58.1%	46.1%	95.7%	80.1%	67.4%	55.3%	94.7%	76.9%	63.8%	52.3%	96.8%	84.0%	72.2%	59.9%
	70%	94.8%	75.6%	60.5%	46.7%	96.6%	82.8%	69.7%	55.9%	95.8%	79.9%	66.3%	52.9%	97.5%	86.3%	74.6%	60.7%
	80%	96.3%	80.3%	64.9%	48.0%	97.6%	86.4%	73.7%	57.2%	97.1%	83.9%	70.2%	54.0%	98.3%	89.4%	78.3%	62.0%
	90%	98.0%	87.6%	73.5%	51.4%	98.8%	91.8%	81.1%	60.7%	98.4%	89.9%	78.1%	57.4%	99.1%	93.7%	85.0%	65.8%
	95%	99.0%	92.8%	82.4%	57.2%	99.4%	95.4%	87.9%	66.7%	99.2%	94.3%	85.6%	62.9%	99.5%	96.5%	90.7%	71.7%
Error % of 1st Approximate Models ($y = C_{s1}$)	10%	-0.3%	-0.9%	-1.9%	-3.2%	-0.2%	-3.5%	-7.0%	-10.9%	-0.1%	-0.8%	-1.8%	-2.6%	-0.2%	-4.3%	-9.8%	-17.4%
	20%	0.0%	-0.8%	-1.8%	-2.9%	-0.1%	-2.9%	-6.5%	-10.7%	-0.1%	-0.7%	-1.7%	-2.3%	-0.1%	-3.9%	-9.5%	-17.3%
	30%	0.0%	-0.6%	-1.6%	-2.7%	-0.1%	-2.5%	-5.9%	-10.2%	-0.1%	-0.8%	-1.4%	-2.3%	-0.1%	-3.3%	-9.2%	-17.2%
	40%	0.1%	-0.7%	-1.4%	-2.5%	-0.1%	-2.2%	-5.1%	-9.8%	0.0%	-0.4%	-1.2%	-2.2%	0.1%	-2.9%	-8.3%	-16.7%
	50%	0.0%	-0.4%	-1.2%	-2.4%	0.0%	-1.6%	-4.5%	-9.1%	-0.1%	-0.3%	-1.3%	-2.1%	0.1%	-2.6%	-7.7%	-16.2%
	60%	0.0%	-0.3%	-1.5%	-2.3%	0.0%	-1.0%	-3.7%	-8.4%	0.0%	-0.2%	-0.9%	-2.2%	0.1%	-2.2%	-6.8%	-15.8%
	70%	0.0%	0.0%	-0.4%	-1.9%	0.0%	-0.6%	-2.6%	-7.9%	0.0%	-0.3%	-0.8%	-1.9%	0.1%	-1.3%	-5.8%	-15.2%
	80%	0.1%	0.1%	-0.3%	-2.2%	0.2%	-0.4%	-1.8%	-6.9%	0.1%	-0.1%	-0.1%	-1.1%	0.2%	-1.1%	-4.1%	-13.7%
	90%	0.0%	0.3%	0.2%	-1.3%	-0.1%	-0.1%	-0.8%	-5.0%	0.1%	0.1%	-0.4%	-1.1%	0.1%	-0.5%	-1.9%	-10.9%
	95%	-0.1%	0.0%	0.0%	-0.9%	0.2%	-0.1%	0.9%	-3.8%	0.1%	0.0%	-0.2%	-0.4%	0.1%	0.0%	-0.5%	-7.4%
Error % of 2nd Approximate Models (y at ρ = 80%)	10%	-1.3%	-2.4%	-2.3%	-1.7%	-4.3%	-10.9%	-10.7%	-6.7%	-0.8%	-2.0%	-2.1%	-1.4%	-7.9%	-19.7%	-19.1%	-11.0%
	20%	-0.8%	-2.3%	-2.2%	-1.3%	-3.9%	-10.0%	-10.1%	-6.5%	-0.8%	-1.9%	-2.0%	-1.1%	-7.2%	-18.5%	-18.5%	-11.0%
	30%	-0.8%	-2.0%	-2.0%	-1.1%	-3.5%	-9.3%	-9.5%	-5.9%	-0.7%	-1.9%	-1.7%	-1.1%	-6.5%	-17.2%	-17.9%	-10.9%
	40%	-0.6%	-1.9%	-1.8%	-0.9%	-3.2%	-8.5%	-8.5%	-5.6%	-0.6%	-1.4%	-1.5%	-1.0%	-5.6%	-15.7%	-16.6%	-10.4%
	50%	-0.6%	-1.5%	-1.6%	-0.8%	-2.7%	-7.3%	-7.8%	-4.9%	-0.6%	-1.2%	-1.6%	-0.8%	-4.9%	-14.2%	-15.5%	-10.1%
	60%	-0.5%	-1.3%	-1.8%	-0.7%	-2.1%	-6.1%	-6.7%	-4.3%	-0.4%	-1.0%	-1.2%	-1.0%	-3.9%	-12.4%	-14.0%	-9.7%
	70%	-0.4%	-0.9%	-0.7%	-0.4%	-1.7%	-4.9%	-5.3%	-3.8%	-0.3%	-1.0%	-1.0%	-0.7%	-3.1%	-9.9%	-12.2%	-9.3%
	80%	-0.2%	-0.6%	-0.6%	-0.8%	-1.0%	-3.6%	-4.0%	-3.0%	-0.1%	-0.6%	-0.3%	0.0%	-2.0%	-7.4%	-9.3%	-8.2%
	90%	-0.2%	-0.1%	0.0%	0.0%	-0.7%	-2.0%	-2.2%	-1.7%	0.0%	-0.2%	-0.5%	-0.1%	-1.0%	-4.1%	-5.2%	-6.2%
	95%	-0.2%	-0.2%	-0.1%	0.1%	-0.2%	-1.1%	0.0%	-1.2%	0.1%	-0.2%	-0.3%	0.3%	-0.5%	-2.0%	-2.5%	-3.9%
Error % of QNA	10%	0.3%	1.9%	2.6%	2.9%	2.2%	7.0%	10.2%	13.2%	0.4%	1.4%	1.7%	2.2%	4.0%	15.0%	23.1%	30.3%
	20%	0.6%	1.8%	2.5%	3.1%	2.0%	6.8%	9.8%	12.5%	0.4%	1.3%	1.7%	2.3%	3.6%	13.8%	21.3%	28.2%
	30%	0.5%	1.7%	2.3%	2.9%	1.6%	6.0%	8.8%	11.6%	0.3%	1.0%	1.6%	2.1%	3.0%	12.1%	18.5%	25.2%
	40%	0.4%	1.3%	2.0%	2.7%	1.2%	4.8%	7.7%	9.8%	0.3%	1.1%	1.5%	1.8%	2.3%	9.6%	15.4%	21.3%
	50%	0.3%	1.1%	1.6%	2.1%	0.8%	3.5%	5.7%	7.9%	0.1%	0.9%	0.9%	1.4%	1.4%	6.3%	10.9%	16.1%
	60%	0.1%	0.6%	0.5%	1.4%	0.3%	1.8%	3.2%	5.1%	0.1%	0.5%	0.7%	0.7%	0.5%	2.4%	5.4%	9.3%
	70%	-0.1%	0.2%	0.6%	0.8%	-0.4%	-0.6%	0.3%	1.4%	-0.1%	-0.2%	0.0%	0.2%	-0.8%	-1.8%	-1.4%	1.1%
	80%	-0.2%	-0.6%	-0.6%	-0.8%	-1.0%	-3.6%	-4.0%	-3.0%	-0.1%	-0.6%	-0.3%	0.0%	-2.0%	-7.4%	-9.3%	-8.2%
	90%	-0.5%	-1.4%	-1.8%	-1.6%	-2.1%	-7.2%	-9.6%	-8.8%	-0.3%	-1.3%	-1.9%	-1.4%	-3.5%	-13.7%	-19.1%	-20.1%
	95%	-0.7%	-2.3%	-3.1%	-2.9%	-2.3%	-9.4%	-12.4%	-13.7%	-0.3%	-1.8%	-2.7%	-2.0%	-4.4%	-17.0%	-25.1%	-28.1%
Error % of QNET	10%	-1.9%	-1.4%	-1.1%	-1.1%	-5.5%	-4.4%	-3.5%	-2.4%	-1.0%	-0.9%	-1.0%	-0.7%	-18.9%	-9.0%	-3.6%	-0.4%
	20%	-1.5%	-1.3%	-1.1%	-0.8%	-5.2%	-4.0%	-3.1%	-2.3%	-1.0%	-0.9%	-0.9%	-0.5%	-18.7%	-9.5%	-3.9%	-0.6%
	30%	-1.5%	-1.1%	-1.0%	-0.6%	-5.1%	-3.8%	-2.8%	-1.8%	-0.9%	-0.9%	-0.7%	-0.5%	-18.4%	-10.0%	-4.2%	-0.6%
	40%	-1.4%	-1.2%	-0.8%	-0.4%	-4.8%	-3.7%	-2.2%	-1.5%	-0.8%	-0.6%	-0.5%	-0.4%	-17.7%	-10.9%	-4.2%	-0.3%
	50%	-1.3%	-1.0%	-0.7%	-0.4%	-4.4%	-3.4%	-2.0%	-0.9%	-0.9%	-0.5%	-0.7%	-0.2%	-16.8%	-12.2%	-4.8%	-0.2%
	60%	-1.2%	-1.0%	-1.1%	-0.3%	-3.9%	-3.2%	-1.7%	-0.4%	-0.7%	-0.5%	-0.4%	-0.4%	-15.3%	-13.8%	-5.6%	-0.3%
	70%	-1.1%	-0.8%	-0.2%	0.0%	-3.3%	-3.3%	-1.3%	-0.2%	-0.6%	-0.7%	-0.4%	-0.2%	-13.3%	-15.1%	-7.4%	-0.5%
	80%	-0.7%	-0.9%	-0.4%	-0.6%	-2.4%	-3.3%	-1.5%	0.2%	-0.3%	-0.5%	0.1%	0.5%	-10.3%	-16.2%	-10.2%	-0.5%
	90%	-0.6%	-0.6%	-0.3%	0.0%	-1.7%	-2.8%	-2.2%	0.5%	-0.2%	-0.3%	-0.4%	0.2%	-6.2%	-13.9%	-14.3%	-2.0%
	95%	-0.4%	-0.6%	-0.6%	-0.3%	-0.7%	-2.1%	-1.2%	-0.5%	0.0%	-0.3%	-0.4%	0.4%	-3.5%	-9.4%	-13.7%	-5.7%

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 0.9, 0.1)				Average Error		Weighted Error	
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	By Util.	Overall	By Util.	Overall
Intrinsic Ratio: (Sim QT – LB) / (UB – LB)	10%	94.9%	96.4%	96.8%	97.2%				
	20%	95.5%	96.2%	96.6%	97.1%				
	30%	94.4%	95.7%	96.5%	96.9%				
	40%	94.0%	95.3%	96.3%	96.8%				
	50%	93.9%	95.0%	95.9%	96.7%				
	60%	92.5%	94.9%	95.4%	96.4%				
	70%	92.2%	94.1%	95.2%	96.1%				
	80%	91.0%	93.7%	94.4%	96.2%				
	90%	89.3%	92.4%	93.4%	95.6%				
	95%	72.6%	93.0%	92.9%	95.1%				
Sim QT of The 1st Server	10%	0.3	1.4	2.2	2.9				
	20%	0.7	2.9	4.7	6.6				
	30%	1.1	4.8	7.9	11.3				
	40%	1.5	6.9	11.9	17.4				
	50%	1.9	9.5	16.9	25.9				
	60%	2.4	12.7	23.8	38.0				
	70%	2.9	16.6	33.2	57.6				
	80%	3.5	21.7	47.5	94.1				
	90%	4.1	28.5	71.4	184.4				
	95%	4.4	32.9	90.5	312.5				
Sim QT of The 2nd Server	10%	1.8	1.8	1.8	1.8				
	20%	4.1	4.0	4.0	3.9				
	30%	7.0	6.9	6.8	6.7				
	40%	10.9	10.7	10.6	10.4				
	50%	16.4	16.0	15.8	15.6				
	60%	24.6	24.1	23.7	23.4				
	70%	38.3	37.5	36.9	36.2				
	80%	65.7	64.6	63.4	62.4				
	90%	148.1	146.3	143.8	140.4				
	95%	312.3	311.2	307.1	298.3				
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	84.8%	56.8%	45.0%	37.3%	64.3%	70.6%		
	20%	85.8%	57.8%	45.5%	37.3%	64.9%			
	30%	86.9%	59.1%	46.2%	37.4%	65.6%			
	40%	88.2%	60.7%	47.1%	37.5%	66.5%			
	50%	89.6%	62.8%	48.3%	37.7%	67.6%			
	60%	91.2%	65.6%	49.9%	38.1%	69.0%			
	70%	93.0%	69.3%	52.6%	38.6%	70.9%			
	80%	95.0%	74.8%	57.1%	39.9%	73.9%			
	90%	97.3%	83.7%	66.8%	43.2%	79.1%			
	95%	98.6%	90.4%	77.2%	48.8%	84.2%			
Error % of 1st Approximate Models ($y = C_{s1}$)	10%	0.0%	-1.2%	-2.4%	-4.0%	4.9%	3.3%	2.6%	1.8%
	20%	-0.1%	-1.0%	-2.1%	-3.7%	4.6%		2.4%	
	30%	0.1%	-0.6%	-1.9%	-3.4%	4.3%		2.3%	
	40%	0.1%	-0.3%	-1.6%	-3.2%	4.0%		2.1%	
	50%	0.1%	-0.1%	-1.1%	-3.0%	3.7%		2.0%	
	60%	0.2%	0.0%	-0.6%	-2.4%	3.3%		1.8%	
	70%	0.2%	0.3%	-0.3%	-1.9%	2.9%		1.6%	
	80%	0.2%	0.4%	0.3%	-1.9%	2.4%		1.4%	
	90%	0.2%	0.5%	0.7%	-1.0%	1.6%		1.1%	
	95%	0.3%	0.2%	0.6%	-0.2%	1.2%		0.9%	
Error % of 2nd Approximate Models (y at ρ = 80%)	10%	-1.3%	-3.3%	-2.9%	-1.8%	9.2%	5.6%	5.9%	3.7%
	20%	-1.3%	-3.0%	-2.7%	-1.5%	8.6%		5.5%	
	30%	-1.0%	-2.5%	-2.5%	-1.2%	7.9%		5.2%	
	40%	-0.8%	-2.1%	-2.2%	-1.0%	7.1%		4.7%	
	50%	-0.7%	-1.7%	-1.7%	-0.8%	6.4%		4.3%	
	60%	-0.5%	-1.5%	-1.1%	-0.3%	5.4%		3.7%	
	70%	-0.3%	-0.9%	-0.7%	0.2%	4.4%		3.1%	
	80%	-0.2%	-0.5%	0.0%	0.0%	3.2%		2.3%	
	90%	0.0%	-0.1%	0.5%	0.8%	2.1%		1.5%	
	95%	0.2%	-0.1%	0.4%	1.1%	1.4%		1.0%	
Error % of QNA	10%	0.9%	2.7%	3.9%	4.6%	18.1%	11.0%	9.3%	6.3%
	20%	0.7%	2.6%	3.8%	4.5%	17.2%		8.9%	
	30%	0.8%	2.6%	3.5%	4.4%	15.6%		8.1%	
	40%	0.6%	2.4%	3.1%	3.9%	13.5%		7.0%	
	50%	0.5%	1.9%	2.7%	3.2%	10.6%		5.5%	
	60%	0.4%	1.2%	2.2%	2.7%	7.1%		3.6%	
	70%	0.1%	0.6%	1.1%	1.8%	3.2%		1.6%	
	80%	-0.2%	-0.5%	0.0%	0.0%	3.2%		2.3%	
	90%	-0.5%	-1.8%	-2.0%	-1.5%	8.9%		6.6%	
	95%	-0.5%	-2.9%	-3.7%	-3.0%	12.9%		10.2%	
Error % of QNET	10%	-3.0%	-2.4%	-1.9%	-1.6%	10.4%	13.1%	7.3%	10.3%
	20%	-3.0%	-2.3%	-1.7%	-1.4%	11.5%		8.5%	
	30%	-2.8%	-2.0%	-1.6%	-1.2%	11.7%		8.7%	
	40%	-2.7%	-1.8%	-1.5%	-1.0%	11.8%		9.0%	
	50%	-2.5%	-1.7%	-1.1%	-0.8%	12.1%		9.4%	
	60%	-2.2%	-1.7%	-0.8%	-0.4%	12.6%		9.9%	
	70%	-2.0%	-1.6%	-0.7%	0.0%	13.2%		10.6%	
	80%	-1.5%	-1.7%	-0.5%	-0.3%	16.3%		12.4%	
	90%	-1.0%	-1.5%	-0.8%	0.1%	15.1%		13.0%	
	95%	-0.4%	-1.3%	-1.2%	0.0%	16.0%		14.0%	

APPENDIX E

Results of STQF when service time SCV is smaller than 1:

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 0.1, 0.1)				Exp-Gam-Gam (1, 0.1, 0.5)				Exp-Gam-Gam (1, 0.5, 0.1)				Exp-Gam-Gam (1, 0.5, 0.5)			
BN Util \ (ST1/ST2)		30/10	30/20	30/25	30/30	30/10	30/20	30/25	30/30	30/10	30/20	30/25	30/30	30/10	30/20	30/25	30/30
Intrinsic Ratio: Sim QT / UB	10%	1.1%	16.3%	28.5%	40.8%	12.1%	43.2%	55.1%	64.5%	38.7%	60.4%	67.5%	73.3%	52.1%	72.6%	78.5%	82.8%
	20%	1.1%	15.7%	27.8%	40.4%	11.6%	42.3%	54.7%	64.3%	37.9%	59.1%	66.4%	72.5%	51.0%	71.8%	77.7%	82.2%
	30%	1.1%	14.9%	27.0%	40.2%	11.2%	41.3%	54.0%	64.3%	37.1%	57.9%	65.4%	71.7%	50.2%	70.5%	76.9%	81.6%
	40%	1.0%	14.2%	26.0%	40.0%	10.8%	40.0%	53.1%	64.1%	36.3%	56.7%	64.3%	70.9%	49.2%	69.5%	76.1%	81.2%
	50%	1.0%	13.3%	24.9%	39.7%	10.3%	38.6%	52.1%	64.0%	35.5%	55.3%	63.0%	70.2%	48.3%	68.1%	75.0%	80.6%
	60%	0.9%	12.4%	23.5%	39.4%	9.9%	36.9%	50.6%	64.0%	34.7%	53.8%	61.8%	69.4%	47.2%	66.8%	73.9%	80.1%
	70%	0.9%	11.4%	21.9%	39.1%	9.4%	35.0%	48.6%	63.8%	33.9%	52.2%	60.2%	69.0%	46.3%	65.3%	72.5%	79.9%
	80%	0.8%	10.3%	19.8%	38.9%	9.0%	32.6%	45.7%	63.8%	33.1%	50.5%	58.2%	68.5%	45.2%	63.4%	70.8%	79.9%
	90%	0.8%	9.2%	16.9%	38.6%	8.5%	29.9%	41.4%	63.9%	32.2%	48.3%	55.4%	68.2%	44.1%	61.3%	68.1%	79.2%
	95%	0.8%	8.6%	15.1%	38.9%	8.3%	28.2%	38.1%	64.4%	31.8%	47.1%	53.3%	69.1%	43.5%	59.8%	65.8%	78.9%
Sim QT of The 1st Server	10%	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
	20%	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6
	30%	7.1	7.1	7.1	7.1	7.1	7.1	7.1	7.1	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.7
	40%	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
	50%	16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5	22.5	22.5	22.5	22.5	22.5	22.5	22.5	22.5
	60%	24.8	24.8	24.7	24.8	24.7	24.8	24.7	24.8	33.7	33.7	33.7	33.7	33.7	33.8	33.7	33.8
	70%	38.5	38.4	38.4	38.5	38.4	38.5	38.4	38.5	52.4	52.5	52.5	52.5	52.5	52.5	52.5	52.5
	80%	65.9	65.9	66.2	65.9	66.0	66.2	66.0	66.1	89.7	90.2	89.9	90.1	90.0	90.3	90.2	90.1
	90%	147.8	148.4	148.6	148.2	147.8	148.3	149.2	148.5	201.8	201.9	202.0	202.1	201.6	201.4	201.9	200.9
	95%	313.2	315.2	312.5	312.0	311.2	310.9	312.6	314.5	427.0	428.4	422.4	433.4	426.9	431.8	426.8	429.5
Sim QT of The 2nd Server	10%	0.0	0.1	0.4	0.7	0.0	0.5	0.9	1.6	0.1	0.5	0.8	1.3	0.1	0.8	1.3	2.1
	20%	0.0	0.3	0.8	1.7	0.1	1.0	2.1	3.6	0.1	1.0	1.8	3.0	0.3	1.7	2.9	4.6
	30%	0.0	0.4	1.2	2.8	0.1	1.6	3.4	6.2	0.2	1.6	3.0	5.1	0.4	2.6	4.8	7.9
	40%	0.0	0.6	1.8	4.4	0.1	2.2	5.0	9.6	0.3	2.3	4.4	7.8	0.6	3.8	7.1	12.2
	50%	0.0	0.7	2.4	6.5	0.2	2.9	7.0	14.4	0.4	3.0	6.2	11.6	0.7	5.1	10.0	18.1
	60%	0.0	0.9	3.2	9.8	0.2	3.7	9.5	21.6	0.5	3.9	8.5	17.2	0.9	6.7	13.9	27.0
	70%	0.0	1.1	4.2	15.1	0.2	4.6	12.8	33.5	0.6	5.0	11.6	26.6	1.1	8.6	19.0	42.0
	80%	0.0	1.3	5.4	25.7	0.2	5.6	17.1	57.4	0.7	6.3	16.0	45.2	1.2	10.9	26.5	71.9
	90%	0.0	1.5	7.0	57.4	0.3	6.7	23.3	129.4	0.8	8.0	22.8	101.3	1.4	13.8	38.3	160.5
	95%	0.0	1.6	7.9	122.0	0.3	7.3	27.1	275.1	0.8	8.9	27.8	216.7	1.5	15.5	46.9	337.4
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	0.12%	6.54%	16.28%	29.00%	1.68%	20.20%	33.86%	46.75%	2.85%	15.93%	25.23%	34.97%	5.11%	23.73%	34.85%	45.29%
	20%	0.10%	6.06%	15.64%	28.80%	1.48%	19.15%	33.18%	46.71%	2.58%	15.11%	24.53%	34.68%	4.63%	22.75%	34.09%	45.12%
	30%	0.09%	5.46%	14.87%	28.67%	1.30%	17.97%	32.27%	46.73%	2.30%	14.19%	23.71%	34.44%	4.16%	21.51%	33.28%	44.89%
	40%	0.08%	4.89%	13.98%	28.57%	1.12%	16.56%	31.15%	46.66%	2.01%	13.12%	22.76%	34.20%	3.65%	20.17%	32.22%	44.79%
	50%	0.06%	4.25%	12.89%	28.39%	0.93%	14.92%	29.71%	46.60%	1.71%	11.91%	21.56%	33.97%	3.11%	18.49%	30.90%	44.63%
	60%	0.05%	3.53%	11.58%	28.26%	0.74%	12.97%	27.73%	46.59%	1.40%	10.48%	20.10%	33.76%	2.56%	16.50%	29.11%	44.46%
	70%	0.04%	2.78%	9.86%	28.15%	0.56%	10.67%	24.93%	46.51%	1.07%	8.73%	18.08%	33.61%	1.98%	14.03%	26.59%	44.44%
	80%	0.03%	1.93%	7.60%	28.03%	0.37%	7.78%	20.62%	46.48%	0.73%	6.57%	15.11%	33.40%	1.35%	10.75%	22.73%	44.36%
	90%	0.01%	1.01%	4.48%	27.91%	0.19%	4.33%	13.51%	46.58%	0.38%	3.80%	10.16%	33.39%	0.70%	6.40%	15.95%	44.41%
	95%	0.01%	0.52%	2.47%	28.11%	0.09%	2.29%	7.98%	46.66%	0.19%	2.05%	6.18%	33.34%	0.35%	3.46%	9.90%	43.99%
Error % of 2nd Approximate Models (y at ρ = 99.9/80%)	10%	1501.7%	12.6%	-35.6%	16.7%	230.3%	-7.1%	-27.2%	-4.4%	41.1%	-9.5%	-19.1%	-3.3%	28.0%	-8.1%	-15.0%	-4.9%
	20%	1562.9%	16.8%	-34.1%	17.8%	246.3%	-5.2%	-26.6%	-4.2%	44.2%	-7.6%	-17.8%	-2.2%	30.8%	-7.1%	-14.1%	-4.3%
	30%	1639.6%	23.4%	-32.0%	18.5%	259.4%	-3.0%	-25.7%	-4.2%	47.4%	-5.6%	-16.4%	-1.1%	32.9%	-5.3%	-13.3%	-3.6%
	40%	1705.4%	29.6%	-29.4%	19.0%	272.9%	0.3%	-24.4%	-3.9%	50.4%	-3.7%	-15.0%	-0.1%	35.6%	-3.9%	-12.3%	-3.1%
	50%	1786.0%	37.9%	-26.2%	20.1%	290.1%	3.9%	-23.0%	-3.7%	53.8%	-1.2%	-13.3%	1.0%	38.3%	-1.9%	-11.0%	-2.5%
	60%	1865.7%	48.3%	-22.1%	20.9%	307.0%	8.8%	-20.7%	-3.7%	57.6%	1.5%	-11.5%	2.2%	41.3%	-0.2%	-9.7%	-1.8%
	70%	1980.2%	60.9%	-16.1%	21.8%	326.9%	14.5%	-17.4%	-3.4%	61.4%	4.6%	-9.2%	2.7%	44.1%	2.2%	-7.9%	-1.6%
	80%	2073.3%	77.4%	-7.2%	22.5%	347.0%	23.1%	-12.2%	-3.5%	65.2%	8.3%	-6.2%	3.6%	47.6%	5.2%	-5.7%	-1.5%
	90%	2186.7%	99.5%	8.5%	23.3%	369.4%	34.4%	-3.2%	-3.6%	69.4%	13.1%	-1.4%	4.0%	51.2%	8.9%	-2.0%	-0.7%
	95%	2253.4%	113.1%	21.2%	22.4%	382.6%	42.4%	5.4%	-4.3%	72.0%	16.0%	2.5%	2.6%	53.3%	11.6%	1.4%	-0.3%
Error % of QNA	10%	8559.5%	508.6%	247.9%	143.1%	718.4%	130.1%	80.4%	54.2%	157.1%	64.9%	47.5%	35.8%	91.2%	37.3%	26.9%	20.4%
	20%	8667.9%	515.9%	247.7%	139.3%	742.5%	130.6%	78.4%	51.8%	159.2%	66.1%	47.8%	35.4%	93.3%	37.3%	27.0%	20.0%
	30%	8684.3%	523.0%	243.6%	130.5%	747.5%	128.8%	75.2%	47.2%	158.7%	65.7%	46.7%	33.8%	93.1%	37.6%	26.1%	18.8%
	40%	8453.0%	514.1%	234.5%	117.1%	740.3%	125.9%	70.4%	41.0%	155.3%	63.4%	44.3%	30.7%	92.3%	36.3%	24.5%	16.6%
	50%	8077.5%	497.9%	220.0%	100.6%	726.5%	120.2%	63.2%	32.8%	149.5%	60.3%	40.7%	26.2%	89.9%	34.7%	22.2%	13.7%
	60%	7458.8%	470.3%	199.6%	79.0%	695.4%	112.5%	54.9%	22.5%	141.2%	55.4%	35.4%	20.5%	86.4%	31.7%	19.1%	9.9%
	70%	6693.3%	425.5%	174.1%	53.1%	651.2%	101.6%	45.3%	10.7%	129.6%	48.8%	29.1%	12.6%	80.6%	28.1%	15.4%	4.7%
	80%	5543.4%	360.5%	140.8%	22.5%	586.3%	89.1%	34.8%	-3.5%	114.5%	40.5%	21.8%	3.6%	73.9%	24.0%	11.1%	-1.5%
	90%	4104.0%	266.8%	99.4%	-12.7%	501.4%	72.2%	24.1%	-19.6%	95.9%	30.8%	14.1%	-7.4%	65.4%	19.2%	7.2%	-7.9%
	95%	3255.7%	203.9%	72.8%	-32.8%	451.5%	62.8%	20.5%	-28.8%	85.6%	25.2%	10.7%	-14.7%	60.6%	16.9%	6.3%	-11.4%
Error % of QNET	10%	1778.3%	83.4%	19.3%	-15.5%	283.0%	29.1%	8.1%	-2.3%	50.4%	5.3%	-2.2%	-6.6%	35.7%	4.1%	-1.0%	-3.5%
	20%	1808.2%	85.2%	21.2%	-14.8%	294.9%	30.3%	8.3%	-2.1%	52.5%	6.9%	-1.0%	-5.6%	37.8%	4.8%	-0.1%	-2.9%
	30%	1853.6%	89.4%	23.5%	-14.3%	302.5%	31.5%	8.8%	-2.1%	54.6%	8.3%	0.1%	-4.5%	39.1%	6.2%	0.4%	-2.2%
	40%	1883.0%	91.4%	25.9%	-13.9%	309.5%	33.5%	9.6%	-1.8%	56.5%	9.5%	1.3%	-3.5%	40.9%	7.1%	1.2%	-1.6%
	50%	1927.1%	94.2%	28.1%	-13.1%	319.8%	35.3%	10.2%	-1.6%	58.6%	11.0%	2.5%	-2.5%	42.5%	8.4%	2.0%	-1.0%
	60%	1968.2%	97.3%	29.8%	-12.5%	329.0%	37.3%	11.3%	-1.6%	61.1%	12.4%	3.5%	-1.4%	44.5%	9.1%	2.7%	-0.2%
	70%	2044.7%	99.9%	31.1%	-11.9%	341.0%	38.7%	12.7%	-1.3%	63.6%	13.6%	4.6%	-0.8%	46.1%	10.0%	3.6%	-0.1%
	80%	2100.5%	103.4%	30.6%	-11.4%	353.7%	40.4%	14.2%	-1.4%	66.3%	14.5%	5.4%	0.0%	48.5%	10.9%	4.2%	0.0%
	90%	2181.8%	110.0%	29.7%	-10.8%	370.4%	41.6%	14.1%	-1.5%	69.6%	15.6%	5.8%	0.4%	51.3%	11.4%	4.5%	0.8%
	95%	2236.9%	115.8%	30.3%	-11.5%	381.8%	44.6%	13.4%	-2.2%	71.8%	16.7%	5.8%	-1.0%	53.2%	12.3%	4.6%	1.2%

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 0.9, 0.5)				Exp-Gam-Gam (1, 0.5, 0.9)				Exp-Gam-Gam (1, 0.9, 0.9)				Exp-Gam-Gam (1, 0.1, 0.9)			
BN Util \ (ST1/ST2)		30/10	30/20	30/25	30/30	30/10	30/20	30/25	30/30	10/30	20/30	25/30	30/30	30/10	30/20	30/25	30/30
Intrinsic Ratio: Sim QT / UB	10%	91.3%	95.0%	96.0%	96.8%	61.4%	79.7%	84.5%	87.6%	93.0%	96.2%	97.3%	97.9%	25.3%	58.4%	68.1%	75.3%
	20%	90.9%	94.8%	95.9%	96.8%	60.2%	78.6%	83.7%	87.4%	92.5%	96.0%	97.2%	97.4%	24.4%	57.6%	67.7%	75.2%
	30%	90.4%	94.6%	95.8%	96.5%	59.1%	77.9%	83.0%	86.7%	92.1%	96.1%	96.8%	97.5%	23.6%	56.6%	67.2%	75.1%
	40%	90.0%	94.3%	95.4%	96.5%	58.1%	76.8%	82.3%	86.5%	91.9%	95.7%	96.6%	97.4%	22.8%	55.2%	66.6%	75.0%
	50%	89.8%	94.0%	95.2%	96.2%	56.9%	75.8%	81.6%	86.0%	91.6%	95.3%	96.7%	97.1%	21.9%	54.0%	65.6%	75.1%
	60%	89.5%	93.7%	95.1%	95.9%	56.0%	74.4%	80.7%	85.9%	91.3%	95.2%	96.2%	97.0%	21.1%	52.2%	64.5%	75.0%
	70%	89.1%	93.4%	94.7%	96.1%	54.7%	73.3%	79.5%	85.5%	91.1%	94.7%	95.9%	97.2%	20.2%	50.2%	62.7%	74.8%
	80%	88.7%	92.7%	94.2%	95.8%	53.6%	71.3%	78.0%	85.2%	90.6%	94.2%	95.7%	96.7%	19.3%	47.4%	60.1%	75.2%
	90%	88.3%	92.4%	93.7%	96.2%	52.5%	69.1%	75.4%	85.4%	90.2%	93.7%	95.0%	96.5%	18.3%	43.9%	55.4%	74.8%
	95%	88.2%	91.9%	93.6%	95.7%	51.9%	67.7%	73.5%	85.1%	90.2%	93.6%	94.8%	97.6%	17.9%	41.8%	51.9%	73.2%
Sim QT of The 1st Server	10%	3.2	3.2	3.2	3.2	2.5	2.5	2.5	2.5	3.2	3.2	3.2	3.2	1.8	1.8	1.8	1.8
	20%	7.1	7.1	7.1	7.1	5.6	5.6	5.6	5.6	7.1	7.1	7.1	7.1	4.1	4.1	4.1	4.1
	30%	12.2	12.2	12.2	12.2	9.7	9.6	9.6	9.6	12.2	12.2	12.2	12.2	7.1	7.1	7.1	7.1
	40%	19.0	18.9	19.0	19.0	15.0	15.0	15.0	15.0	19.0	19.0	19.0	19.0	11.0	11.0	11.0	11.0
	50%	28.5	28.5	28.5	28.6	22.5	22.5	22.4	22.5	28.5	28.5	28.5	28.4	16.5	16.5	16.5	16.5
	60%	42.7	42.6	42.8	42.8	33.7	33.8	33.8	33.8	42.8	42.8	42.7	42.7	24.7	24.7	24.7	24.8
	70%	66.4	66.5	66.5	66.6	52.5	52.6	52.5	52.6	66.5	66.4	66.5	66.8	38.5	38.4	38.5	38.5
	80%	113.9	114.1	113.8	114.2	90.2	89.9	90.1	89.8	114.3	113.6	113.9	114.0	66.1	65.9	66.0	66.2
	90%	257.0	256.6	258.0	256.5	202.2	201.0	202.4	203.1	256.4	257.9	255.8	257.8	148.6	148.4	148.4	148.3
	95%	546.6	543.2	556.3	538.5	426.1	427.0	430.5	428.6	542.2	538.2	543.9	539.3	316.1	316.1	310.8	310.6
Sim QT of The 2nd Server	10%	0.2	1.0	1.6	2.4	0.2	1.1	1.8	2.8	0.3	1.3	2.1	3.1	0.1	0.8	1.5	2.4
	20%	0.5	2.2	3.6	5.4	0.4	2.3	4.0	6.2	0.6	2.8	4.6	6.9	0.2	1.7	3.2	5.4
	30%	0.8	3.5	6.0	9.3	0.6	3.7	6.6	10.6	1.0	4.6	7.7	11.9	0.2	2.7	5.3	9.2
	40%	1.0	5.1	8.9	14.5	0.8	5.3	9.8	16.4	1.3	6.6	11.5	18.5	0.3	3.8	7.9	14.2
	50%	1.3	7.1	12.8	21.7	1.1	7.2	13.8	24.5	1.7	9.1	16.4	27.7	0.4	5.1	11.1	21.4
	60%	1.7	9.4	17.8	32.4	1.3	9.4	19.2	36.7	2.2	12.1	22.9	41.5	0.5	6.6	15.3	32.1
	70%	2.0	12.3	24.8	50.5	1.6	12.2	26.4	56.9	2.6	15.8	31.9	64.7	0.6	8.3	20.8	49.8
	80%	2.4	15.9	35.3	86.2	1.9	15.5	37.1	97.1	3.1	20.5	45.4	110.2	0.7	10.3	28.6	85.7
	90%	2.8	20.8	52.7	194.7	2.1	19.7	53.7	218.9	3.7	26.7	67.7	247.4	0.7	12.5	39.4	192.0
	95%	3.1	23.8	66.7	409.1	2.3	22.2	66.3	460.9	4.0	30.7	85.5	528.3	0.8	13.7	46.8	396.5
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	6.93%	24.34%	34.08%	43.42%	7.44%	30.19%	42.19%	52.62%	8.78%	29.20%	39.88%	49.46%	4.33%	30.20%	44.51%	56.57%
	20%	6.39%	23.50%	33.54%	43.29%	6.77%	29.05%	41.39%	52.51%	8.09%	28.26%	39.32%	49.33%	3.85%	28.97%	43.79%	56.50%
	30%	5.81%	22.49%	32.87%	43.23%	6.07%	27.74%	40.55%	52.35%	7.38%	27.22%	38.55%	49.39%	3.40%	27.51%	42.92%	56.50%
	40%	5.19%	21.35%	31.97%	43.25%	5.35%	26.12%	39.45%	52.27%	6.61%	25.78%	37.59%	49.35%	2.94%	25.76%	41.79%	56.43%
	50%	4.51%	19.84%	30.93%	43.10%	4.58%	24.27%	38.14%	52.14%	5.75%	24.10%	36.50%	49.32%	2.45%	23.72%	40.28%	56.44%
	60%	3.78%	18.01%	29.41%	43.06%	3.80%	21.81%	36.19%	52.09%	4.83%	21.99%	34.86%	49.26%	1.98%	21.10%	38.25%	56.41%
	70%	2.97%	15.56%	27.21%	43.12%	2.93%	18.83%	33.47%	51.95%	3.81%	19.17%	32.39%	49.18%	1.49%	17.84%	35.11%	56.37%
	80%	2.08%	12.23%	23.69%	43.01%	2.01%	14.69%	29.15%	51.98%	2.67%	15.26%	28.53%	49.15%	1.00%	13.50%	30.21%	56.41%
	90%	1.09%	7.49%	16.97%	43.16%	1.05%	8.92%	20.98%	51.88%	1.41%	9.39%	20.93%	48.98%	0.50%	7.78%	21.00%	56.41%
	95%	0.56%	4.20%	10.70%	43.17%	0.53%	4.94%	13.35%	51.81%	0.73%	5.40%	13.59%	49.48%	0.25%	4.16%	13.09%	56.08%
Error % of 2nd Approximate Models (y at ρ = 99.9/80%)	10%	2.2%	-1.7%	-2.7%	-1.1%	20.0%	-7.4%	-12.7%	-5.1%	1.8%	-1.5%	-2.6%	-1.3%	108.5%	-9.8%	-22.6%	-7.5%
	20%	2.7%	-1.5%	-2.7%	-1.1%	22.5%	-6.2%	-11.9%	-4.8%	2.5%	-1.3%	-2.5%	-0.8%	116.4%	-8.5%	-22.1%	-7.3%
	30%	3.3%	-1.3%	-2.6%	-0.8%	24.8%	-5.4%	-11.2%	-4.1%	2.9%	-1.4%	-2.1%	-0.9%	123.6%	-6.8%	-21.5%	-7.3%
	40%	3.7%	-1.0%	-2.2%	-0.8%	26.9%	-4.0%	-10.4%	-3.9%	3.1%	-1.0%	-1.9%	-0.8%	131.6%	-4.5%	-20.8%	-7.0%
	50%	4.0%	-0.7%	-1.9%	-0.5%	29.5%	-2.7%	-9.6%	-3.3%	3.4%	-0.6%	-2.0%	-0.4%	141.1%	-2.3%	-19.6%	-7.2%
	60%	4.3%	-0.3%	-1.8%	-0.2%	31.7%	-0.9%	-8.6%	-3.1%	3.7%	-0.4%	-1.6%	-0.4%	150.2%	1.0%	-18.3%	-7.1%
	70%	4.7%	0.0%	-1.4%	-0.4%	34.7%	0.5%	-7.2%	-2.7%	4.1%	0.0%	-1.2%	-0.6%	161.3%	5.1%	-15.9%	-6.9%
	80%	5.3%	0.7%	-0.9%	-0.1%	37.6%	3.4%	-5.5%	-2.4%	4.6%	0.6%	-1.0%	0.0%	173.6%	11.3%	-12.3%	-7.3%
	90%	5.7%	1.0%	-0.4%	-0.4%	40.4%	6.8%	-2.2%	-2.6%	5.0%	1.1%	-0.3%	0.2%	188.5%	20.1%	-4.8%	-6.9%
	95%	5.9%	1.6%	-0.2%	0.1%	42.1%	8.9%	0.4%	-2.3%	5.1%	1.2%	0.0%	-1.0%	194.8%	26.1%	1.6%	-4.8%
Error % of QNA	10%	9.4%	5.2%	4.2%	3.2%	62.4%	25.2%	18.1%	13.9%	7.4%	3.9%	2.7%	2.1%	293.6%	70.3%	46.2%	32.2%
	20%	9.7%	5.2%	4.0%	3.0%	64.3%	25.8%	18.2%	13.3%	7.9%	4.0%	2.7%	2.4%	302.6%	70.3%	44.9%	30.5%
	30%	10.0%	5.1%	3.8%	3.0%	65.2%	25.3%	17.6%	12.6%	8.0%	3.5%	2.9%	2.1%	306.0%	69.3%	42.6%	27.4%
	40%	9.9%	4.9%	3.7%	2.5%	64.8%	24.7%	16.3%	10.7%	7.9%	3.6%	2.6%	1.8%	306.0%	67.4%	38.8%	23.3%
	50%	9.5%	4.6%	3.3%	2.2%	64.1%	23.3%	14.5%	8.6%	7.7%	3.6%	2.1%	1.7%	303.1%	63.3%	34.4%	17.4%
	60%	9.0%	4.2%	2.6%	1.8%	61.7%	21.7%	12.2%	5.4%	7.4%	3.1%	1.9%	1.1%	293.6%	58.9%	28.5%	10.6%
	70%	8.5%	3.6%	2.2%	0.6%	59.2%	18.8%	9.6%	1.9%	7.0%	2.8%	1.6%	0.2%	280.5%	53.0%	22.5%	2.6%
	80%	8.0%	3.2%	1.6%	-0.1%	55.2%	16.7%	6.6%	-2.4%	6.6%	2.6%	1.0%	0.0%	261.6%	47.0%	15.9%	-7.3%
	90%	7.1%	2.4%	0.9%	-1.6%	49.8%	13.9%	4.3%	-7.8%	6.1%	2.2%	0.7%	-0.8%	237.3%	40.4%	11.3%	-17.6%
	95%	6.6%	2.3%	0.4%	-1.8%	47.0%	12.7%	3.8%	-10.4%	5.6%	1.8%	0.5%	-2.4%	220.1%	36.9%	10.3%	-21.8%
Error % of QNET	10%	3.1%	0.1%	-0.4%	-0.9%	26.8%	3.0%	-0.6%	-2.2%	2.7%	0.2%	-0.6%	-0.9%	138.5%	17.2%	5.9%	0.2%
	20%	3.5%	0.3%	-0.4%	-0.9%	28.6%	4.0%	0.1%	-1.9%	3.3%	0.3%	-0.5%	-0.4%	144.2%	17.8%	6.0%	0.4%
	30%	4.1%	0.5%	-0.4%	-0.6%	30.3%	4.5%	0.6%	-1.2%	3.6%	0.1%	-0.1%	-0.5%	148.7%	18.8%	6.2%	0.5%
	40%	4.3%	0.6%	0.0%	-0.6%	31.6%	5.5%	1.1%	-1.0%	3.7%	0.5%	0.0%	-0.4%	153.4%	20.2%	6.2%	0.7%
	50%	4.5%	0.8%	0.1%	-0.3%	33.5%	6.2%	1.5%	-0.4%	3.9%	0.8%	-0.2%	0.0%	159.0%	21.0%	6.7%	0.5%
	60%	4.7%	1.0%	0.1%	0.0%	34.6%	7.2%	2.1%	-0.2%	4.1%	0.8%	0.2%	0.0%	163.7%	22.6%	6.9%	0.7%
	70%	5.0%	1.1%	0.4%	-0.2%	36.6%	7.5%	2.7%	0.2%	4.3%	1.1%	0.4%	-0.2%	170.0%	24.0%	7.7%	0.9%
	80%	5.4%	1.5%	0.6%	0.1%	38.5%	8.7%	3.1%	0.5%	4.7%	1.4%	0.4%	0.4%	177.8%	25.6%	8.6%	0.4%
	90%	5.7%	1.4%	0.6%	-0.2%	40.5%	9.1%	3.7%	0.4%	5.0%	1.5%	0.6%	0.6%	189.3%	26.5%	10.2%	0.9%
	95%	5.9%	1.7%	0.3%	0.2%	42.1%	9.7%	3.4%	0.6%	5.0%	1.3%	0.4%	-0.6%	194.6%	28.0%	9.2%	3.1%

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 0.9, 0.1)				Average Error		Weighted Error	
BN Util \ (ST1/ST2)		30/10	30/20	30/25	30/30	By Util.	Overall	By Util.	Overall
Intrinsic Ratio: Sim QT / UB	10%	88.3%	92.9%	94.4%	95.4%				
	20%	87.9%	92.6%	94.1%	95.2%				
	30%	87.3%	92.3%	93.7%	94.8%				
	40%	87.1%	91.9%	93.5%	94.7%				
	50%	86.8%	91.6%	93.1%	94.3%				
	60%	86.4%	91.2%	92.8%	94.1%				
	70%	86.1%	90.8%	92.5%	94.0%				
	80%	85.8%	90.1%	92.0%	94.1%				
	90%	85.4%	89.6%	91.2%	93.5%				
	95%	85.2%	89.3%	90.8%	93.1%				
Sim QT of The 1st Server	10%	3.2	3.2	3.2	3.2				
	20%	7.1	7.1	7.1	7.1				
	30%	12.2	12.2	12.2	12.2				
	40%	19.0	19.0	19.0	19.0				
	50%	28.5	28.6	28.5	28.5				
	60%	42.8	42.7	42.7	42.7				
	70%	66.4	66.5	66.6	66.4				
	80%	114.7	114.2	114.0	114.2				
	90%	255.8	256.8	256.3	256.8				
	95%	542.4	534.2	545.8	537.0				
Sim QT of The 2nd Server	10%	0.2	0.7	1.2	1.7				
	20%	0.3	1.6	2.6	3.9				
	30%	0.5	2.5	4.3	6.7				
	40%	0.7	3.7	6.4	10.4				
	50%	1.0	5.0	9.1	15.6				
	60%	1.2	6.7	12.8	23.3				
	70%	1.4	8.7	17.8	36.2				
	80%	1.7	11.3	25.3	62.1				
	90%	2.0	14.8	37.6	138.8				
	95%	2.2	17.0	47.5	291.8				
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	5.02%	18.73%	27.15%	35.54%	25.9%	21.6%		
	20%	4.62%	18.07%	26.60%	35.48%	25.4%			
	30%	4.18%	17.21%	26.03%	35.47%	24.8%			
	40%	3.74%	16.20%	25.26%	35.39%	24.1%			
	50%	3.24%	14.99%	24.25%	35.32%	23.3%			
	60%	2.70%	13.53%	23.00%	35.31%	22.3%			
	70%	2.13%	11.61%	21.10%	35.26%	20.9%			
	80%	1.47%	9.02%	18.16%	35.21%	19.1%			
	90%	0.78%	5.45%	12.80%	35.08%	16.3%			
	95%	0.40%	3.08%	8.00%	35.21%	14.1%			
Error % of 2nd Approximate Models (y at ρ = 99.9/80%)	10%	3.0%	-2.1%	-3.7%	-1.2%	60.6%	74.3%	2.5%	1.6%
	20%	3.5%	-1.8%	-3.4%	-1.0%	63.0%		2.3%	
	30%	4.2%	-1.5%	-3.0%	-0.7%	65.6%		2.1%	
	40%	4.3%	-1.0%	-2.7%	-0.5%	67.8%		1.9%	
	50%	4.8%	-0.7%	-2.3%	-0.1%	70.9%		1.7%	
	60%	5.2%	-0.2%	-2.0%	0.1%	74.0%		1.5%	
	70%	5.6%	0.2%	-1.7%	0.2%	78.6%		1.3%	
	80%	6.0%	0.9%	-1.2%	0.1%	82.9%		1.1%	
	90%	6.5%	1.4%	-0.3%	0.8%	88.2%		0.8%	
	95%	6.7%	1.8%	0.1%	1.2%	92.0%		0.6%	
Error % of QNA	10%	13.2%	7.5%	5.9%	4.8%	321.0%	261.9%	11.2%	6.4%
	20%	13.4%	7.6%	5.9%	4.7%	325.0%		10.7%	
	30%	13.6%	7.4%	5.9%	4.6%	324.9%		9.9%	
	40%	13.1%	7.3%	5.4%	4.1%	316.1%		8.8%	
	50%	12.6%	6.7%	5.0%	3.6%	301.9%		7.4%	
	60%	11.9%	6.1%	4.2%	2.8%	278.9%		5.8%	
	70%	10.9%	5.3%	3.3%	1.7%	250.0%		3.9%	
	80%	9.8%	4.5%	2.4%	0.1%	209.0%		2.4%	
	90%	8.5%	3.3%	1.6%	-0.9%	160.2%		1.8%	
	95%	7.7%	2.8%	1.1%	-1.4%	132.2%		1.9%	
Error % of QNET	10%	4.0%	0.0%	-1.0%	-1.6%	70.5%	80.1%	1.7%	1.2%
	20%	4.4%	0.2%	-0.8%	-1.4%	72.1%		1.6%	
	30%	4.9%	0.5%	-0.4%	-1.0%	74.1%		1.5%	
	40%	5.0%	0.8%	-0.3%	-0.8%	75.8%		1.5%	
	50%	5.3%	1.0%	0.1%	-0.5%	77.9%		1.4%	
	60%	5.6%	1.3%	0.2%	-0.3%	79.9%		1.3%	
	70%	5.8%	1.4%	0.3%	-0.1%	83.2%		1.1%	
	80%	6.1%	1.8%	0.5%	-0.2%	85.9%		0.9%	
	90%	6.5%	1.8%	0.7%	0.4%	89.6%		0.7%	
	95%	6.7%	1.9%	0.6%	0.8%	92.2%		0.5%	

APPENDIX F

Results of STQB when service time SCV is greater than 1:

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 2, 2)				Exp-Gam-Gam (1, 2, 5)				Exp-Gam-Gam (1, 5, 2)				Exp-Gam-Gam (1, 5, 5)			
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30
(Sim QT – LB) / (UB – LB)	10%	115.4%	112.8%	111.8%	110.6%	110.4%	109.4%	109.0%	108.8%	130.7%	120.7%	118.0%	116.5%	121.8%	116.4%	114.6%	114.1%
	20%	119.1%	115.1%	113.0%	112.0%	113.5%	110.3%	110.2%	108.9%	133.6%	125.3%	122.4%	120.4%	126.3%	120.5%	118.8%	117.6%
	30%	120.7%	116.9%	114.5%	113.5%	116.1%	111.6%	111.1%	110.3%	139.3%	131.0%	127.0%	124.4%	128.9%	125.0%	122.7%	121.0%
	40%	122.1%	118.2%	116.0%	114.7%	118.0%	113.0%	112.5%	111.3%	146.5%	136.4%	131.6%	128.1%	134.7%	129.1%	126.6%	124.7%
	50%	127.7%	120.2%	117.8%	115.6%	114.9%	114.2%	113.4%	112.3%	150.8%	143.1%	137.0%	132.0%	140.1%	133.2%	130.2%	128.0%
	60%	128.3%	122.6%	119.7%	117.2%	112.8%	117.5%	115.1%	113.6%	160.9%	149.9%	142.4%	135.7%	145.8%	137.9%	135.0%	131.4%
	70%	129.3%	125.4%	122.1%	118.2%	124.3%	116.7%	116.5%	114.8%	172.3%	159.6%	148.9%	139.5%	143.0%	144.8%	139.5%	134.5%
	80%	137.7%	127.3%	124.2%	119.9%	104.8%	120.4%	118.2%	116.2%	188.4%	171.0%	156.7%	142.6%	157.1%	154.4%	147.1%	136.9%
	90%	154.0%	135.9%	128.2%	120.9%	147.8%	125.5%	123.0%	117.3%	177.6%	187.2%	169.4%	147.0%	174.1%	160.0%	155.1%	140.9%
	95%	264.5%	157.1%	131.7%	122.6%	-38.7%	114.1%	123.8%	116.7%	296.1%	213.1%	180.9%	153.5%	103.3%	159.2%	170.6%	147.3%
Sim QT of The 1st Server	10%	0.5	2.1	3.4	4.7	0.5	2.1	3.4	4.7	1.0	4.3	6.8	9.3	1.0	4.3	6.8	9.3
	20%	1.1	4.6	7.5	10.4	1.1	4.6	7.5	10.4	2.1	9.2	15.0	20.8	2.1	9.2	15.0	20.8
	30%	1.7	7.5	12.5	17.8	1.7	7.5	12.5	17.8	3.3	15.0	25.0	35.5	3.3	15.0	25.0	35.5
	40%	2.3	10.9	18.8	27.4	2.3	10.9	18.8	27.5	4.6	21.8	37.5	54.7	4.6	21.8	37.4	54.9
	50%	3.0	15.0	26.7	40.6	3.0	15.0	26.8	40.7	6.0	30.1	53.7	81.4	6.0	30.0	53.5	81.0
	60%	3.7	20.1	37.5	60.1	3.7	20.0	37.5	60.0	7.5	40.0	74.9	120.2	7.5	40.0	75.1	120.5
	70%	4.6	26.2	52.7	91.2	4.6	26.2	52.5	91.3	9.1	52.6	104.8	182.7	9.1	52.5	105.0	182.4
	80%	6.6	34.4	75.0	148.6	5.5	34.3	74.9	148.4	10.9	68.7	149.6	296.9	10.9	68.5	150.6	298.2
	90%	7.8	45.0	112.5	289.4	6.4	45.0	112.4	290.8	12.8	90.0	224.4	583.8	12.8	90.1	224.3	587.8
	95%	8.5	51.9	142.4	492.1	7.0	51.9	142.7	493.2	13.9	103.5	285.0	994.3	13.9	103.5	284.0	982.1
Sim QT of The 2nd Server	10%	5.1	5.3	5.4	5.5	10.1	10.2	10.3	10.4	5.3	5.9	6.2	6.5	10.2	10.7	11.0	11.3
	20%	11.5	11.9	12.2	12.5	22.6	23.0	23.3	23.4	12.0	13.6	14.6	15.5	23.1	24.4	25.3	26.2
	30%	19.6	20.6	21.1	21.7	38.8	39.4	40.0	40.4	20.6	23.9	26.0	27.9	39.5	42.3	44.2	46.0
	40%	30.5	32.0	33.0	34.0	60.4	61.4	62.3	63.1	32.1	37.9	41.8	45.4	61.6	66.3	70.0	73.6
	50%	45.8	48.0	49.8	51.4	90.4	92.1	93.6	95.0	48.0	57.9	64.8	71.1	92.4	100.0	106.2	112.8
	60%	68.6	72.0	74.9	77.8	135.5	138.5	140.7	143.2	72.1	87.5	99.3	110.4	138.4	150.1	161.3	172.8
	70%	106.3	111.7	116.6	121.6	211.1	214.4	218.7	223.4	111.6	136.3	156.3	177.0	213.9	233.5	251.5	272.8
	80%	182.1	189.4	198.2	209.5	360.3	367.0	373.6	384.1	189.6	228.7	265.1	306.6	366.2	397.3	430.7	469.6
	90%	408.5	421.1	436.8	465.8	813.1	821.5	835.9	860.3	415.0	483.5	561.3	678.9	819.5	864.0	934.0	1047.9
	95%	866.4	884.6	900.1	965.5	1700.4	1717.3	1743.9	1791.8	882.3	972.2	1085.7	1378.4	1710.5	1771.4	1911.2	2172.9
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	90.8%	71.2%	61.3%	54.1%	95.1%	82.6%	75.2%	69.1%	83.7%	57.9%	47.8%	41.2%	90.8%	71.5%	61.7%	54.9%
	20%	91.4%	72.1%	62.0%	54.6%	95.5%	83.3%	75.6%	69.2%	84.8%	59.5%	49.3%	42.7%	91.5%	72.6%	62.8%	55.7%
	30%	92.2%	73.3%	62.8%	54.9%	95.9%	84.0%	76.2%	69.5%	86.1%	61.5%	51.0%	44.0%	92.2%	73.8%	63.9%	56.4%
	40%	93.0%	74.6%	63.7%	55.4%	96.3%	84.9%	76.9%	69.7%	87.4%	63.5%	52.8%	45.4%	93.0%	75.3%	65.1%	57.3%
	50%	93.9%	76.2%	65.1%	55.8%	96.8%	86.0%	77.8%	70.0%	88.9%	65.8%	54.7%	46.6%	93.9%	76.9%	66.5%	58.2%
	60%	94.8%	78.2%	66.6%	56.4%	97.3%	87.4%	78.9%	70.5%	90.6%	68.6%	57.0%	47.9%	94.9%	79.0%	68.2%	58.9%
	70%	95.9%	81.0%	68.9%	57.1%	97.9%	89.1%	80.7%	71.0%	92.4%	72.2%	59.9%	49.2%	95.9%	81.6%	70.5%	59.9%
	80%	96.5%	84.6%	72.6%	58.5%	98.5%	91.5%	83.3%	72.1%	94.6%	76.9%	63.9%	50.8%	97.1%	85.3%	74.1%	61.2%
	90%	98.1%	90.3%	79.5%	61.7%	99.2%	94.8%	88.2%	74.7%	97.0%	84.3%	71.4%	53.8%	98.5%	90.6%	80.6%	64.1%
	95%	99.0%	94.5%	86.3%	66.2%	99.6%	97.1%	92.4%	78.4%	98.4%	90.4%	79.2%	58.1%	99.2%	94.5%	87.1%	68.9%
Error % of 1st Approximate Models ($y = C_{s1}$)	10%	2.6%	11.6%	18.7%	26.1%	1.6%	6.7%	10.7%	14.6%	18.1%	74.9%	115.6%	152.4%	10.3%	42.9%	67.6%	90.1%
	20%	2.1%	10.1%	17.4%	24.5%	1.3%	6.2%	10.1%	14.5%	16.1%	66.8%	103.9%	138.8%	9.0%	39.0%	62.1%	84.4%
	30%	1.8%	9.0%	15.9%	22.9%	1.1%	5.7%	9.5%	13.7%	13.7%	58.0%	92.7%	126.2%	8.0%	35.0%	57.1%	79.1%
	40%	1.5%	7.9%	14.5%	21.5%	0.9%	5.0%	8.7%	13.1%	11.1%	50.1%	82.5%	115.3%	6.7%	31.1%	52.0%	73.8%
	50%	0.9%	6.6%	12.7%	20.5%	0.9%	4.4%	8.0%	12.5%	9.1%	41.7%	71.6%	104.8%	5.4%	27.1%	47.1%	69.0%
	60%	0.7%	5.2%	10.9%	18.7%	0.8%	3.5%	7.0%	11.7%	6.5%	33.7%	61.4%	95.7%	4.2%	22.8%	41.2%	64.1%
	70%	0.5%	3.8%	8.7%	17.4%	0.4%	3.0%	6.0%	10.9%	4.2%	24.6%	50.2%	86.5%	3.4%	17.7%	35.1%	59.5%
	80%	0.1%	2.6%	6.5%	15.3%	0.6%	2.0%	4.7%	9.7%	2.0%	15.8%	37.8%	78.4%	2.0%	12.0%	26.6%	54.8%
	90%	-0.2%	0.6%	3.4%	12.8%	-0.1%	0.9%	2.5%	8.2%	1.4%	6.8%	21.7%	65.7%	0.8%	6.6%	16.5%	46.0%
	95%	-1.0%	-0.9%	1.5%	9.5%	0.7%	0.8%	1.4%	6.7%	-1.1%	1.1%	11.2%	49.8%	1.0%	3.8%	7.9%	34.4%
Error % of 2nd Approximate Models (y at $\rho = 80\%$)	10%	6.4%	15.0%	15.0%	11.5%	3.5%	8.5%	8.8%	6.9%	24.5%	57.3%	58.2%	45.3%	13.6%	33.3%	35.1%	28.2%
	20%	5.5%	13.4%	13.8%	10.1%	3.1%	7.9%	8.2%	6.8%	22.0%	50.5%	50.0%	37.6%	12.1%	29.9%	31.0%	24.5%
	30%	4.9%	12.0%	12.5%	8.8%	2.7%	7.3%	7.6%	6.1%	19.0%	42.9%	42.3%	30.5%	10.8%	26.4%	27.4%	21.1%
	40%	4.2%	10.8%	11.1%	7.6%	2.3%	6.5%	6.9%	5.6%	15.8%	36.3%	35.4%	24.4%	9.1%	23.2%	23.9%	17.6%
	50%	3.3%	9.2%	9.6%	6.8%	2.1%	5.8%	6.3%	5.1%	13.2%	29.2%	28.2%	18.7%	7.6%	19.9%	20.6%	14.7%
	60%	2.7%	7.6%	7.9%	5.4%	1.8%	4.7%	5.5%	4.4%	9.9%	22.7%	21.7%	13.8%	6.0%	16.4%	16.8%	11.8%
	70%	2.1%	5.7%	6.0%	4.5%	1.2%	4.1%	4.6%	3.8%	6.9%	15.4%	14.9%	9.1%	4.8%	12.3%	13.2%	9.3%
	80%	1.2%	4.1%	4.3%	3.0%	1.1%	2.7%	3.5%	3.1%	3.9%	8.6%	8.1%	5.5%	3.0%	7.8%	8.4%	7.2%
	90%	0.4%	1.5%	1.9%	2.0%	0.2%	1.3%	1.7%	2.3%	2.4%	2.3%	0.7%	1.1%	1.3%	4.1%	3.9%	4.2%
	95%	-0.7%	-0.4%	0.6%	0.8%	0.9%	1.1%	1.0%	2.0%	-0.6%	-1.4%	-2.6%	-3.7%	1.2%	2.4%	0.1%	0.5%
Error % of QNA	10%	-1.5%	-5.1%	-7.2%	-8.7%	-0.5%	-1.9%	-2.9%	-3.8%	-5.8%	-14.6%	-19.0%	-22.6%	-2.1%	-6.3%	-8.6%	-11.1%
	20%	-1.6%	-5.3%	-7.1%	-8.9%	-0.6%	-1.8%	-2.8%	-3.4%	-5.5%	-15.2%	-20.2%	-23.8%	-2.2%	-6.7%	-9.5%	-11.9%
	30%	-1.4%	-4.9%	-6.7%	-8.6%	-0.5%	-1.6%	-2.5%	-3.2%	-5.1%	-15.1%	-19.8%	-23.2%	-1.8%	-6.4%	-9.2%	-11.5%
	40%	-1.1%	-4.0%	-5.7%	-7.5%	-0.4%	-1.2%	-2.0%	-2.5%	-4.5%	-13.4%	-17.7%	-20.8%	-1.4%	-5.3%	-7.9%	-10.3%
	50%	-0.9%	-2.8%	-4.3%	-5.5%	0.0%	-0.5%	-1.0%	-1.6%	-2.9%	-10.8%	-14.5%	-17.0%	-0.8%	-3.3%	-5.4%	-7.8%
	60%	-0.2%	-1.3%	-2.4%	-3.5%	0.3%	0.1%	0.0%	-0.4%	-1.3%	-6.4%	-9.4%	-11.4%	0.1%	-0.5%	-2.3%	-4.3%
	70%	0.5%	0.9%	0.2%	-0.5%	0.4%	1.5%	1.5%	1.2%	0.9%	-0.6%	-2.3%	-4.4%	1.7%	3.0%	2.5%	0.5%
	80%	1.2%	4.1%	4.3%	3.0%	1.1%	2.7%	3.5%	3.1%	3.9%	8.6%	8.1%	5.5%	3.0%	7.8%	8.4%	7.2%
	90%	2.1%	7.7%	10.1%	8.9%	1.1%	4.5%	6.0%	6.0%	9.3%	24.0%	26.3%	19.9%	4.8%	16.2%	19.2%	16.3%
	95%	2.0%	9.6%	14.8%	13.5%	2.2%	6.2%	8.3%	8.8%	9.9%	35.0%	44.6%	31.8%	6.7%	22.3%	26.9%	22.9%
Error % of QNET	10%	6.2%	7.6%	7.2%	6.9%	1.7%	2.9%	-0.8%	2.8%	33.7%	38.4%	37.5%	35.3%	12.1%	17.7%	18.7%	18.2%
	20%	5.5%	6.5%	6.4%	5.6%	1.4%	2.6%	2.6%	2.8%	31.9%	33.9%	31.2%	28.3%	10.9%	15.5%	15.6%	14.8%
	30%	5.0%	5.7%	5.4%	4.4%	1.1%	2.2%	2.2%	2.1%	29.4%	28.8%	25.5%	21.9%	10.0%	13.2%	13.0%	11.8%
	40%	4.6%	5.1%	4.6%	3.4%	0.9%	1.8%	1.8%	1.7%	26.8%	24.8%	20.6%	16.4%	8.8%	11.3%	10.5%	8.8%
	50%	3.8%	4.3%	3.6%	2.7%	0.9%	1.4%	1.4%	1.3%	24.9%	20.5%	15.7%	11.4%	7.7%	9.4%	8.4%	6.2%
	60%	3.4%	3.5%	2.6%	1.5%	0.9%	0.7%	0.9%	0.7%	22.1%	17.1%	11.7%	7.2%	6.6%	7.5%	6.0%	3.8%
	70%	3.0%	2.7%	1.7%	0.8%	0.5%	0.6%	0.5%	0.2%	19.4%	13.2%	8.0%	3.5%	6.0%	5.3%	4.1%	1.9%
	80%	2.2%	2.1%	1.2%	-0.1%	0.7%	0.1%	0.0%	-0.3%	16.2%	10.4%	5.3%	1.4%	4.5%	3.0%	1.6%	0.7%
	90%	1.3%	1.0%	0.5%	-0.1%	0.1%	-0.2%	-0.8%	-0.5%	12.7%	9.2%	3.9%	0.2%	2.8%	2.5%	0.1%	-0.5%
	95%	-0.1%	-0.1%	0.4%	-0.1%	0.8%	0.3%	-0.5%	-0.2%	6.4%	7.7%	5.0%	-0.5%	2.3%	2.6%	-1.1%	-2.1%

(C_{a1}^2, C_{s1}^2, C_{s2}^2)		Exp-Gam-Gam (1, 8, 5)				Exp-Gam-Gam (1, 5, 8)				Exp-Gam-Gam (1, 8, 8)				Exp-Gam-Gam (1, 2, 8)			
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30	10/30	20/30	25/30	29/30
(Sim QT – LB) / (UB – LB)	10%	121.2%	117.1%	116.1%	114.3%	115.5%	114.0%	112.9%	112.2%	118.7%	114.9%	113.9%	112.5%	121.3%	107.7%	107.0%	106.7%
	20%	128.2%	122.6%	120.7%	119.5%	119.0%	117.1%	116.4%	114.9%	122.3%	119.7%	118.5%	117.9%	116.3%	109.3%	109.5%	107.8%
	30%	132.7%	128.9%	126.1%	124.3%	122.4%	120.5%	119.9%	118.8%	129.8%	124.2%	123.4%	121.9%	121.1%	110.0%	110.7%	109.0%
	40%	138.4%	134.6%	131.3%	129.2%	132.3%	125.2%	122.7%	121.3%	131.5%	131.7%	128.8%	126.7%	117.4%	109.8%	109.2%	109.7%
	50%	147.3%	142.6%	137.7%	134.2%	122.7%	129.9%	127.0%	124.2%	137.5%	135.7%	133.9%	130.7%	135.9%	112.3%	109.8%	110.6%
	60%	155.8%	151.3%	144.1%	137.7%	128.5%	130.5%	131.6%	126.1%	141.1%	143.5%	139.6%	135.1%	152.2%	110.5%	112.8%	111.1%
	70%	162.4%	159.8%	151.0%	142.4%	145.6%	138.1%	133.5%	130.4%	146.4%	153.6%	144.7%	140.0%	155.6%	113.2%	115.4%	112.8%
	80%	168.9%	172.6%	160.1%	145.6%	127.5%	141.1%	136.6%	132.5%	158.7%	155.9%	150.7%	142.6%	111.9%	122.7%	115.7%	112.4%
	90%	197.1%	194.5%	174.6%	152.5%	76.0%	143.6%	150.5%	133.9%	174.2%	184.0%	160.5%	145.1%	18.6%	126.4%	116.1%	116.5%
	95%	172.5%	220.6%	185.7%	159.7%	100.5%	145.7%	164.5%	135.6%	193.3%	214.6%	171.8%	144.4%	93.6%	110.0%	105.8%	114.3%
Sim QT of The 1st Server	10%	1.5	6.4	10.2	14.0	1.0	4.3	6.8	9.3	1.5	6.4	10.2	14.0	0.5	2.1	3.4	4.7
	20%	3.2	13.8	22.5	31.3	2.1	9.2	15.0	20.9	3.2	13.8	22.5	31.3	1.1	4.6	7.5	10.4
	30%	5.0	22.5	37.5	53.3	3.3	15.0	25.0	35.5	5.0	22.5	37.5	53.3	1.7	7.5	12.5	17.8
	40%	6.9	32.7	56.1	82.5	4.6	21.9	37.5	55.0	6.9	32.9	56.3	82.2	2.3	10.9	18.7	27.4
	50%	9.0	45.0	80.5	122.9	6.0	30.0	53.6	81.5	9.0	45.0	80.3	121.4	3.0	15.0	26.8	40.7
	60%	11.2	60.1	112.3	180.2	7.5	40.0	75.0	119.6	11.2	60.1	111.8	180.1	3.7	20.0	37.4	60.1
	70%	13.7	79.1	157.8	272.7	9.1	52.4	105.2	182.6	13.7	78.9	157.5	275.1	4.6	26.2	52.5	91.1
	80%	16.4	102.9	224.9	443.9	10.9	68.6	149.6	294.9	16.3	102.8	225.5	445.7	5.5	34.3	74.9	148.0
	90%	19.3	135.2	336.1	881.9	12.9	90.1	224.6	581.0	19.3	134.6	338.3	875.8	6.4	45.1	112.6	291.9
	95%	20.9	155.0	428.8	1485.1	13.9	103.6	284.3	991.4	20.8	155.9	428.2	1447.7	7.0	51.8	142.5	486.0
Sim QT of The 2nd Server	10%	10.3	11.1	11.6	12.0	15.2	15.6	15.9	16.1	15.3	16.0	16.4	16.8	15.1	15.2	15.2	15.3
	20%	23.4	25.6	27.2	28.6	34.2	35.3	36.2	36.8	34.5	36.5	37.9	39.4	33.9	34.2	34.5	34.6
	30%	40.2	45.1	48.4	51.5	58.6	60.9	62.8	64.5	59.3	63.3	66.6	69.5	58.2	58.6	59.2	59.4
	40%	62.7	71.3	77.6	84.0	91.5	95.5	98.5	101.7	92.2	100.4	106.2	111.9	90.4	91.1	91.7	92.7
	50%	94.3	109.2	120.3	131.7	136.4	144.0	149.5	154.7	138.4	151.1	162.2	172.4	136.1	136.8	137.6	139.3
	60%	141.3	165.8	184.6	203.0	204.6	214.7	226.2	233.8	207.1	228.6	247.1	265.8	204.5	204.6	207.3	209.2
	70%	218.5	257.1	290.3	325.8	319.2	335.0	350.2	370.3	321.4	357.2	385.4	424.3	317.5	318.5	323.1	326.7
	80%	371.3	434.7	495.2	563.1	543.0	568.2	595.0	636.6	549.6	597.5	654.0	729.7	540.6	547.8	551.7	558.4
	90%	828.7	937.6	1061.9	1268.2	1211.9	1254.3	1328.5	1412.3	1229.3	1328.4	1419.2	1608.9	1209.8	1226.9	1233.1	1263.1
	95%	1725.1	1897.5	2076.6	2585.9	2565.1	2612.3	2748.8	2913.7	2584.5	2743.1	2872.1	3216.6	2564.6	2570.2	2573.2	2634.8
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	87.0%	63.3%	53.3%	46.2%	93.6%	78.5%	70.0%	63.4%	90.8%	71.3%	61.6%	54.5%	96.7%	87.6%	81.7%	76.7%
	20%	87.9%	64.9%	54.7%	47.8%	94.1%	79.3%	70.7%	63.8%	91.5%	72.5%	62.8%	55.7%	96.9%	88.1%	82.1%	76.8%
	30%	88.9%	66.7%	56.3%	49.1%	94.6%	80.3%	71.6%	64.5%	92.2%	73.7%	64.0%	56.6%	97.2%	88.6%	82.6%	77.0%
	40%	90.0%	68.6%	58.0%	50.5%	95.2%	81.3%	72.4%	64.9%	93.0%	75.3%	65.3%	57.7%	97.5%	89.3%	83.0%	77.2%
	50%	91.3%	70.8%	59.9%	51.7%	95.8%	82.7%	73.6%	65.5%	93.9%	77.0%	66.9%	58.7%	97.8%	90.1%	83.7%	77.4%
	60%	92.6%	73.4%	62.2%	53.0%	96.5%	84.3%	75.1%	66.2%	94.9%	79.2%	68.8%	59.6%	98.2%	91.1%	84.7%	77.7%
	70%	94.1%	76.5%	64.8%	54.4%	97.2%	86.5%	76.9%	67.0%	95.9%	81.9%	71.0%	60.7%	98.6%	92.4%	86.0%	78.2%
	80%	95.8%	80.9%	68.8%	55.9%	98.0%	89.2%	79.9%	68.3%	97.1%	85.3%	74.4%	62.1%	99.0%	94.1%	88.0%	79.0%
	90%	97.7%	87.4%	76.0%	59.0%	99.0%	93.3%	85.5%	70.9%	98.5%	90.8%	80.8%	64.8%	99.5%	96.5%	91.6%	81.2%
	95%	98.8%	92.4%	82.9%	63.5%	99.5%	96.2%	90.6%	74.6%	99.2%	94.6%	87.0%	69.0%	99.7%	98.0%	94.8%	84.4%
Error % of 1st Approximate Models ($y = C_{s1}$)	10%	24.3%	96.0%	146.5%	196.0%	7.4%	30.1%	47.6%	64.3%	16.7%	67.7%	105.2%	142.0%	0.7%	4.8%	7.7%	10.5%
	20%	21.2%	86.6%	134.3%	178.6%	6.6%	27.8%	44.4%	61.5%	15.0%	61.9%	97.5%	131.0%	0.8%	4.3%	6.9%	10.1%
	30%	18.7%	76.9%	121.5%	163.9%	5.8%	25.4%	41.3%	57.7%	12.9%	56.4%	89.7%	123.4%	0.6%	4.0%	6.5%	9.7%
	40%	16.0%	68.0%	109.8%	150.5%	4.6%	22.5%	38.4%	55.2%	11.4%	49.3%	81.6%	114.8%	0.6%	3.8%	6.6%	9.4%
	50%	12.9%	57.8%	97.0%	137.8%	4.4%	19.5%	34.6%	52.3%	9.5%	43.8%	73.8%	107.7%	0.1%	3.2%	6.1%	9.0%
	60%	10.1%	47.6%	84.5%	128.9%	3.5%	17.3%	30.5%	50.1%	7.7%	36.6%	65.2%	100.2%	-0.2%	3.0%	5.2%	8.7%
	70%	7.6%	37.7%	71.6%	117.8%	2.2%	13.4%	27.0%	45.8%	5.8%	28.5%	56.4%	91.9%	-0.2%	2.3%	4.2%	8.0%
	80%	5.0%	26.1%	55.8%	108.5%	1.9%	10.0%	21.9%	42.5%	3.7%	21.9%	45.5%	85.6%	0.3%	1.2%	3.5%	7.7%
	90%	2.0%	12.7%	34.4%	89.8%	1.6%	5.7%	12.4%	37.0%	1.7%	10.0%	29.1%	74.8%	0.7%	0.6%	2.3%	5.7%
	95%	1.3%	5.1%	20.0%	69.9%	0.7%	3.1%	6.1%	29.5%	0.7%	3.9%	16.5%	63.2%	0.1%	0.6%	2.0%	5.0%
Error % of 2nd Approximate Models (y at $\rho = 80\%$)	10%	24.2%	57.4%	58.8%	48.9%	9.6%	23.5%	25.0%	20.9%	16.6%	40.8%	43.0%	36.6%	1.9%	6.0%	6.4%	5.3%
	20%	21.2%	50.5%	51.5%	40.3%	8.6%	21.6%	22.6%	19.0%	14.9%	36.6%	38.3%	30.6%	2.0%	5.5%	5.7%	4.9%
	30%	18.6%	43.6%	44.1%	33.2%	7.6%	19.5%	20.4%	16.3%	12.9%	32.7%	33.5%	26.5%	1.6%	5.1%	5.3%	4.6%
	40%	15.9%	37.4%	37.4%	26.7%	6.3%	17.0%	18.4%	14.6%	11.3%	27.5%	28.7%	21.9%	1.6%	4.8%	5.4%	4.3%
	50%	12.9%	30.3%	30.2%	20.6%	5.9%	14.5%	15.8%	12.7%	9.4%	24.0%	24.3%	18.2%	0.9%	4.1%	5.0%	4.0%
	60%	10.1%	23.5%	23.7%	16.6%	4.7%	12.9%	13.1%	11.4%	7.7%	19.1%	19.7%	14.4%	0.5%	3.8%	4.1%	3.8%
	70%	7.5%	17.2%	17.4%	11.8%	3.2%	9.6%	11.3%	8.8%	5.8%	13.8%	15.6%	10.5%	0.3%	3.0%	3.3%	3.2%
	80%	5.0%	10.3%	10.4%	8.5%	2.6%	7.1%	8.7%	7.4%	3.7%	10.4%	11.1%	8.4%	0.7%	1.7%	2.7%	3.1%
	90%	2.0%	3.1%	2.6%	2.7%	1.9%	4.0%	3.5%	6.0%	1.7%	3.3%	5.3%	6.1%	0.8%	0.9%	1.8%	1.8%
	95%	1.3%	-0.4%	-0.6%	-1.9%	0.8%	2.1%	0.7%	4.3%	0.7%	0.1%	1.7%	5.5%	0.2%	0.8%	1.6%	1.8%
Error % of QNA	10%	-3.1%	-9.4%	-13.4%	-15.8%	-1.0%	-3.7%	-5.2%	-6.6%	-1.8%	-5.7%	-8.2%	-9.8%	-0.7%	-1.0%	-1.5%	-1.9%
	20%	-3.4%	-10.4%	-14.5%	-17.9%	-1.0%	-3.7%	-5.7%	-6.9%	-1.7%	-6.2%	-9.0%	-11.8%	-0.5%	-1.1%	-1.8%	-2.0%
	30%	-2.9%	-10.4%	-14.4%	-17.8%	-0.8%	-3.4%	-5.4%	-7.0%	-1.8%	-5.8%	-9.0%	-11.3%	-0.5%	-0.8%	-1.6%	-1.8%
	40%	-2.3%	-8.9%	-12.7%	-16.1%	-0.9%	-2.8%	-4.1%	-5.6%	-1.0%	-5.4%	-7.9%	-10.2%	-0.2%	-0.4%	-0.7%	-1.3%
	50%	-1.4%	-6.9%	-10.0%	-13.1%	0.2%	-1.6%	-2.7%	-3.7%	-0.3%	-2.9%	-5.5%	-7.5%	-0.5%	-0.1%	0.0%	-0.6%
	60%	0.0%	-3.4%	-5.6%	-7.4%	0.7%	1.0%	-0.5%	-0.5%	0.8%	-0.4%	-2.1%	-3.9%	-0.5%	0.7%	0.4%	0.4%
	70%	2.2%	2.4%	1.1%	-1.1%	1.1%	3.1%	3.6%	2.4%	2.2%	3.1%	3.4%	0.7%	-0.2%	1.3%	1.2%	1.3%
	80%	5.0%	10.3%	10.4%	8.5%	2.6%	7.1%	8.7%	7.4%	3.7%	10.4%	11.1%	8.4%	0.7%	1.7%	2.7%	3.1%
	90%	8.0%	22.7%	26.3%	20.3%	4.3%	12.4%	14.3%	15.0%	5.8%	17.1%	23.1%	20.0%	1.4%	3.0%	4.7%	4.3%
	95%	10.7%	32.3%	42.6%	31.2%	4.5%	15.7%	19.3%	21.0%	7.0%	22.7%	32.8%	32.0%	1.1%	4.2%	6.6%	6.5%
Error % of QNET	10%	27.3%	35.6%	35.7%	36.0%	6.1%	10.4%	11.5%	11.7%	14.4%	22.1%	23.5%	24.4%	0.2%	1.5%	1.8%	1.8%
	20%	24.8%	31.0%	30.3%	28.3%	5.4%	9.1%	9.7%	10.0%	13.2%	19.3%	19.9%	19.0%	0.3%	1.1%	1.2%	1.5%
	30%	22.8%	26.3%	24.7%	21.9%	4.7%	7.8%	8.0%	7.6%	11.6%	16.8%	16.4%	15.4%	0.2%	1.0%	0.9%	1.2%
	40%	20.8%	22.5%	20.0%	16.2%	3.8%	6.2%	6.7%	6.2%	10.6%	13.3%	12.9%	11.3%	0.2%	0.9%	1.1%	0.9%
	50%	18.4%	18.1%	15.0%	10.9%	3.8%	4.7%	4.9%	4.5%	9.3%	11.4%	10.0%	8.2%	-0.2%	0.5%	0.9%	0.7%
	60%	16.3%	14.0%	10.9%	7.6%	3.2%	4.1%	3.2%	3.6%	8.2%	8.4%	7.1%	5.0%	-0.5%	0.5%	0.3%	0.5%
	70%	14.3%	10.8%	7.4%	3.8%	2.2%	2.2%	2.4%	1.4%	6.9%	5.3%	4.9%	1.9%	-0.4%	0.1%	-0.2%	0.0%
	80%	11.9%	7.5%	4.1%	1.9%	2.2%	1.4%	1.4%	0.6%	5.3%	4.5%	2.9%	0.8%	0.2%	-0.6%	-0.3%	0.1%
	90%	7.8%	5.1%	1.4%	-1.0%	1.9%	1.0%	-1.5%	0.4%	3.3%	1.2%	0.7%	0.5%	0.6%	-0.5%	-0.4%	-0.9%
	95%	5.3%	4.0%	2.1%	-2.1%	0.9%	0.8%	-2.1%	0.1%	1.9%	0.2%	0.0%	2.2%	0.1%	0.1%	0.3%	-0.4%

$(C_{a1}^2, C_{s1}^2, C_{s2}^2)$		Exp-Gam-Gam (1, 8, 2)				Average Error		Weighted Error	
BN Util \ (ST1/ST2)		10/30	20/30	25/30	29/30	By Util.	Overall	By Util.	Overall
(Sim QT – LB) / (UB – LB)	10%	129.8%	120.5%	118.0%	116.4%				
	20%	137.4%	127.0%	123.8%	121.7%				
	30%	144.4%	133.9%	129.9%	127.0%				
	40%	152.9%	141.7%	136.3%	132.2%				
	50%	163.7%	150.1%	142.9%	137.3%				
	60%	177.8%	160.7%	150.6%	142.8%				
	70%	191.8%	173.5%	159.2%	146.9%				
	80%	211.0%	191.7%	170.8%	151.5%				
	90%	252.1%	216.5%	188.7%	157.5%				
	95%	282.0%	242.6%	211.1%	167.6%				
Sim QT of The 1st Server	10%	1.5	6.4	10.2	13.9				
	20%	3.2	13.8	22.5	31.3				
	30%	5.0	22.4	37.5	53.3				
	40%	6.9	32.6	56.2	82.2				
	50%	9.0	45.0	80.3	121.8				
	60%	11.3	60.2	112.7	180.6				
	70%	13.7	78.7	157.8	272.9				
	80%	16.4	102.4	225.5	444.8				
	90%	19.3	135.2	335.3	868.8				
	95%	20.8	155.3	428.3	1483.2				
Sim QT of The 2nd Server	10%	5.5	6.3	6.8	7.3				
	20%	12.5	15.0	16.6	18.0				
	30%	21.5	26.9	30.5	33.7				
	40%	33.7	43.6	50.4	56.5				
	50%	50.7	67.6	79.5	90.6				
	60%	76.3	103.9	124.4	144.5				
	70%	117.6	162.9	198.2	233.1				
	80%	198.2	274.3	339.3	409.3				
	90%	434.3	562.2	704.3	907.0				
	95%	893.0	1076.6	1330.0	1847.6				
Sim QT_2 / (Sim QT_1 + Sim QT_2)	10%	77.9%	49.5%	40.2%	34.4%	69.1%	75.7%		
	20%	79.4%	52.0%	42.5%	36.6%	70.1%			
	30%	81.1%	54.5%	44.9%	38.7%	71.1%			
	40%	82.9%	57.2%	47.3%	40.7%	72.3%			
	50%	84.9%	60.0%	49.7%	42.6%	73.5%			
	60%	87.1%	63.3%	52.5%	44.5%	75.0%			
	70%	89.6%	67.4%	55.7%	46.1%	76.8%			
	80%	92.4%	72.8%	60.1%	47.9%	79.2%			
	90%	95.7%	80.6%	67.7%	51.1%	83.2%			
	95%	97.7%	87.4%	75.6%	55.5%	87.0%			
Error % of 1st Approximate Models ($y = C_{s1}$)	10%	43.5%	165.1%	246.5%	318.8%	66.8%	40.4%	35.4%	23.5%
	20%	37.5%	144.0%	215.5%	279.4%	60.3%		32.9%	
	30%	32.2%	124.6%	188.0%	246.6%	54.3%		30.5%	
	40%	26.7%	105.9%	163.5%	219.6%	48.7%		28.1%	
	50%	21.1%	88.4%	141.4%	196.1%	43.3%		25.7%	
	60%	15.5%	70.6%	119.6%	174.7%	38.0%		23.1%	
	70%	10.6%	52.9%	98.3%	159.2%	32.6%		20.3%	
	80%	5.9%	34.2%	74.3%	142.8%	26.9%		17.2%	
	90%	1.4%	15.9%	45.1%	120.7%	19.4%		12.8%	
	95%	0.0%	5.8%	23.1%	91.5%	13.4%		9.3%	
Error % of 2nd Approximate Models (y at ρ = 80%)	10%	43.4%	97.3%	97.2%	76.7%	30.8%	14.6%	18.1%	9.4%
	20%	37.4%	82.4%	80.2%	60.2%	26.7%		16.2%	
	30%	32.1%	68.8%	65.2%	46.5%	22.8%		14.3%	
	40%	26.7%	55.9%	52.1%	35.4%	19.2%		12.4%	
	50%	21.1%	44.0%	40.5%	25.7%	15.7%		10.5%	
	60%	15.4%	32.0%	29.3%	17.0%	12.3%		8.5%	
	70%	10.6%	20.6%	18.9%	11.1%	8.9%		6.4%	
	80%	5.9%	9.2%	8.1%	5.3%	5.7%		4.3%	
	90%	1.4%	-0.1%	-2.7%	-1.0%	2.3%		1.9%	
	95%	0.0%	-3.8%	-9.1%	-8.9%	1.8%		1.4%	
Error % of QNA	10%	-8.2%	-20.1%	-25.7%	-29.9%	8.2%	8.3%	4.5%	5.6%
	20%	-8.7%	-21.8%	-27.9%	-32.2%	8.7%		4.9%	
	30%	-8.2%	-21.6%	-27.6%	-31.5%	8.5%		4.9%	
	40%	-7.2%	-19.8%	-25.1%	-28.3%	7.4%		4.3%	
	50%	-5.6%	-16.1%	-20.5%	-23.2%	5.6%		3.3%	
	60%	-3.2%	-10.8%	-14.1%	-16.6%	3.3%		1.9%	
	70%	0.7%	-2.8%	-4.9%	-6.8%	1.9%		1.4%	
	80%	5.9%	9.2%	8.1%	5.3%	5.7%		4.3%	
	90%	12.8%	32.5%	33.0%	23.5%	13.5%		10.5%	
	95%	18.2%	53.7%	58.3%	37.3%	20.1%		16.4%	
Error % of QNET	10%	66.7%	71.2%	68.0%	64.1%	21.7%	10.3%	12.9%	6.9%
	20%	61.7%	60.7%	54.8%	49.1%	18.5%		11.4%	
	30%	57.3%	51.4%	43.4%	36.7%	15.5%		9.9%	
	40%	52.8%	42.9%	33.8%	26.6%	12.8%		8.6%	
	50%	48.2%	35.5%	25.7%	18.0%	10.4%		7.2%	
	60%	43.4%	28.5%	18.4%	10.5%	8.1%		5.9%	
	70%	39.2%	22.5%	12.7%	5.9%	6.0%		4.7%	
	80%	34.2%	17.4%	7.9%	2.1%	4.4%		3.6%	
	90%	25.2%	16.7%	6.4%	0.2%	3.1%		2.8%	
	95%	17.2%	17.3%	7.4%	-1.9%	2.7%		2.4%	

REFERENCES

Adan, I. and J. Resing. 2001. Queueing Theory. Lecture Notes. <http://www.win.tue.nl/~iadan/queueing.pdf>

Ankenman, B., B. L. Nelson, and J. Statum. 2008. Stochastic Kriging for Simulation Metamodeling. Winter Sim. Conf. 362–370.

Avi-Itzhak, B. and P. Naor. 1963. Some Queueing Problems with the Service Station subject to Breakdown. Operations Research. 11: 303–320.

Avi-Itzhak, B. 1965. A sequence of Service Stations with Arbitrary Input and Regular Service Times. Management Science. 11(5): 565–571.

Ashcroft, H. 1950. The Productivity of Several Machines under the Care of One Operator. Journal of the Royal Statistical Society. Series B (Methodological). 12(1): 145–151.

Asmussen, S. 1992. Queueing Simulation in Heavy Traffic. Math. Oper. Res. 17(1): 84–111.

Asmussen, S. and Glynn, P. W. 2000. Stochastic Simulation. New York: Springer.

Bailey, N. T. 1954. On Queueing Processes with Bulk Service. Journal of the Royal Statistical Society Series B (Methodological). 16(1): 80–87.

Bard, Y. 1979. Some Extensions to Multiclass Queueing Network Analysis. Performance of Computer Systems. Proc of the Int Symp on Modelling and Perform Eval of Comput Syst: 51–62.

Baskett, F., K. M. Chandy, R. M. Richard and G. P. Fernando. 1975. Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. J. ACM. 22(2): 248–260.

Berry, A. C. 1941. The Accuracy of the Gaussian Approximation to the Sum of Independent Variables. Trans. Amer. Math. Soc. 49: 122–136.

Billings, R. 2006. FabSim: A Discrete-Event Simulation Model for Wafer Fabs. <http://www2.isye.gatech.edu/~rbilling/courses/isye4803/Project/FabSim.pdf>

Bitran, G. R. and D. Tirupati. 1989a. Approximations for Product Departures from a Single-Server Station with Batch Processing in Multi-Product Queues. Management Science. 35(7): 851–878.

Bitran, G. R. and D. Tirupati. 1989b. Tradeoff Curves, Targeting and Balancing in Manufacturing Queueing Networks. Operations Research. 37(4): 547–564.

Boebel, F. G. and O. Ruelle. 1996. Cycle Time Reduction Program at ACL. Proc. IEEE/ SEMI ASMC: 12–14.

Bruell, S. C. and G. Balbo. 1980. Computational Algorithms for Closed Queueing Networks. New York: North Holland.

Burke, P. J. 1956. The Output of a Queuing System. Operations Research. 4(6): 699–704.

Buzacott, J. A., and L. E. Hanifin. 1978. Models of Automatic Transfer Lines with Inventory Banks – A review and comparison. AIIE Trans. 10(2): 197–207.

Buzacott, J. A., and G. Shanthikumar. 1993. Stochastic Models of Manufacturing Systems. New Jersey: Pren-tice-Hall.

Chandy K. M., U. Herzog, L. Woo. 1975. Parametric Analysis of Queuing Networks. IBM Journal of Research and Development. 19(1): 36–42.

Chandy K. M., U. Herzog, L. Woo. 1975. Approximate Analysis of General Queueing Networks. IBM Journal of Research and Development. 19(1): 43–49.

Chaudhry, M. L. and J. G. C. Templeton. 1983. A First Course in Bulk Queues. New York: Wiley.

Chen, H. and Yao, D. D. 2001. Fundamentals of Queueing Networks: Performance, Asymptotics and Optimization. New York: Springer.

Cobham, A. 1954. Priority Assignment in Waiting Line Problems. Journal of the Operations Research Society of America. 2(1): 70–76.

Collins, D. W., K. Williams, and F. C. Hoppensteadt. 1997. Implementation of Minimum Inventory Variability Scheduling 1-Step Ahead Policy in a Large Semiconductor Manufacturing Facility. Proc. 6th ETFA Int'l Conf: 497–504.

Cusumano, M. A. 1988. Manufacturing Innovation: Lessons from the Japanese Auto Industry. Sloan Management Review. Fall 1988: 29–39.

Dallery, Y. and D. D. Kouvatsos. 1998. Queueing networks with blocking. Ann. Oper. Res. 79.

Dai, J. 1992. Performance Analysis of Queueing Networks Using Reflecting Brownian Motions. http://www2.isye.gatech.edu/people/faculty/dai/Software.html

Dai, J. G. and J. M. Harrison. 1991. Steady-State Analysis of RBM in a Rectangle: Numerical Methods and a Queueing Application. The Annals of Applied Probability. 1(1): 16–35.

Dai, J. G. and J. M. Harrison. 1992. Reflected Brownian Motion in an Orthant: Numerical Methods for Steady-State Analysis. The Annals of Applied Probability. 2: 65–86.

Dai, J. G., V. Nguyen and M. I. Reiman. 1994. Sequential Bottleneck Decomposition: An Approximation Method for Generalized Jackson Networks. Operations Research. 42(1): 119–136.

Daley, D. J. 1968. The Correlation Structure of the Output Process of Some Single Server Queueing Systems. The Annals of Mathematical Statistics. 38(3): 1007–1019.

DNS. 2009. http://www.dsninnovations.org/aboutus/default.aspx

Equipment Engineering Capabilities Guidelines. 2002. International SEMATECH.

Erlang, A. K. 1909. The Theory of Probabilities and Telephone Conversations. Nyt Tidsskrift for Matematik B. 20: 33–39.

Esseen, C. G. 1945. Fourier analysis of distribution functions: A Mathematical Study of the Laplace–Gaussian Law. Acta Math. 77(1): 1–125.

Fowler, J. W., S. Brown, H, Gold, and A. Schoeming, 1997. Measurable Improvements in Cycle-Time-Constrained Capacity. Proc. IEEE ISSM: A21–A24.

Friedman, H. D. 1965. Reduction Methods for Tandem Queueing Systems. Operations Research. 13(1): 121–131.

Fowler, J. W., N. Phojanamongkolkij, J. K. Cochran and D. C. Montgomery. 2002. Optimal Batching in a Wafer Fabrication Facility Using a Multiproduct G/G/c Model with Batch Processing. International Journal of Production Research. 40(2): 275–292.

Gaver, D. P. 1962. A Waiting Line with Interrupted Service, including Priorities. J. Royal Statist. Soc. Series B. 24: 73–90.

Gordon, W. J. and Newell, G. F. 1967. Closed Queueing Systems with Exponential Servers. Operations Research. 15(2): 254–265.

Gross, D., and C. M. Harris. 1998. Queueing Theory. New York: Wiley.

Gupta, U. C. and Rao, T. S. S. S. V. 1994. A Recursive Method to Compute the Steady State Probabilities of the Machine Interference Model: (M/G/1)/K. Computers & Operations Research. 21(6): 597–605.

Gupta, U. C. and Rao, T. S. S. S. V. 1996. On the M/G/1 Machine Interference Model with Spares. European Journal of Operational Research. 89(1): 164–171.

Harrison, J. M. and A. Nguyen. 1990. The QNET Method for Two-Moment Analysis of Open Queueing Networks. Queueing Systems. 6: 1–32.

Harrison, J. M. and R. J. Williams. 1992. Brownian Models of Feedforwd Queueing Networks: Quasireversibility and Product Form Solutions. The Annals of Applied Probability. 2(2): 263–293.

Hayes, R. H., S. C. Wheelwright and K. B. Clark. 1988. Dynamic Manufacturing: Creating the Learning Organization. New York: The Free Press.

Haque, L. and M. J. Armstrong. 2007. A Survey of the Machine Interference Problem. European Journal of Operational Research. 179(2): 469–482.

Heyman, D. P. 1975. A Diffusion Model Approximation for the GI/G/1 Queue in Heavy Traffic. Bell System Technical Journal. 54(9): 1637–1646.

Hopp, W. J. 2008. Management Science and the Science of Management. Management Science. 54(12): 1961–1962.

Hopp, W. J. and M. L. Spearman. 1996. Factory Physics. Chicago, IL: IRWIN.

Iglehart, D. L. and W. Whitt. 1970. Multiple Channel Queues in Heavy Traffic, II: Sequences, Networks, and Batches. Advances in Applied Probability. 2(2):355–369.

Jackson, J. R. 1957. Networks of Waiting Lines. Operations Research. 5(4): 518–521.

Jaiswal, N. K. 1968. Priority Queues. New York: Academic Press.

Kraemer, W. and M. Lagenbach-Belz. 1976. Approximate Formulae for the Delay in the Queueing System GI/G/1. Eighth Int. Teletraffic Congress, Melbourne. 235.1–235.8.

Keilson, J. 1962. Queues subject to Service Interruption. Ann. Math. Statist. 33: 1314–1322.

Kendall, D. G. 1952. Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. Ann. Math. Statist. 24(3): 338–354.

Khintchine, A. Y. 1932. Mathematisches uber die Erwortung vor einemoffenthchen Schalter. Mat. Sb. 39: 73–84.

Khintchine, A. Y. 1933. On Mean Time of Stoppage of Machines. Mathematiceski Sbornic. 40: 119–123.

Kingman, J. F. C. 1962a. Some Inequalities for the Queue GI/G/1. Biometrika. 49: 315–324.

Kingman, J. F. C. 1962b. On Queues in Heavy Traffic. Journal of the Royal Statistical Society. Series B (Methodological). 24(2): 383-392.

Kingman, J. F. C. 1965. The Heavy Traffic Approximation in the Theory of Queues. Proc. Symp. on Congestion Theory. University of North Caroline Press: 137–159.

Kleinrock, L. 1976. Queueing Systems: Computer Applications. New York: Wiley-Interscience.

Kobayashi, H. 1974. Application of the Diffusion Approximation to Queueing Networks I: Equilibrium Queue Distributions. J. ACM. 21(2): 316–328.

Kobayashi, H. 1974. Application of the Diffusion Approximation to Queueing Networks II: Nonequilibrium Distributions and Applications to Computer Modeling. J. ACM. 21(3): 459–469.

Kraemer, W. and M. Langenbach-Belz. 1976. Approximations Formulae for the Delay in the Queueing System GI/G/1. Proc. Eighth International Teletraffic Congress, Melbourne: 235, 1–8.

Kuehn, P. J. 1979. Approximate Analysis of General Queuing Networks by Decomposition. IEEE Transactions on Communications. 27(1): 113–126.

Law, A. M., and W. D. Kelton. 2000. Simulation modeling & analysis. 3rd ed. New York: McGraw-Hill, Inc.

Leachman, R. C. and D. A. Hodges. 1996. Benching Semiconductor Manufacturing. IEEE Transactions on Semiconductor Manufacturing. 9(2): 158–169.

Lindley, D. V. 1952. Theory of Queues with a Single Server. Proc. Comb. Phil. Soc. 48: 277–289.

Little, J. D. C. 1961. A Proof of the Queueing Formula: L=W. Operations Research. 9: 383–387.

Marshall, K. T. 1968. Some Inequalities in Queueing. Operations Research. 16(3): 651–665.

Mack, C., T. Murphy, and N. L. Webb. 1957. The Efficiency of N Machines Uni-Directionally Patrolled by One Operative when Walking Time and Repair Times

are Constants. Journal of the Royal Statistical Society. Series B (Methodological). 19(1): 166–172.

Miller, R. G., Jr. 1960. Priority Queues. Ann. Math. Statist. 31: 86–103.

Morrison, J. R. and Donald P. Martin. 2007. Practical Extensions to Cycle Time Approximations for the G/G/m-Queue With Applications. IEEE Trans. Auto. Sci. Eng. 4(4): 523–532.

Nazzal, D. and M. Mollaghasemi. 2001. Critical Tools Identification and Characteristics Curves Construction in a Wafer Fabrication Facility. Proc. Winter Sim. Conf. 2: 1194–1199.

Newell, G. F. 1979. Approximate Behavior of Tandem Queues. New York: Springer.

Niu, S. C. 1980. Bounds for the Expected Delays in Some Tandem Queues. Journal of Applied Probability. 17(3): 831–838.

Palm, D. C. 1958. The Assignment of Workers in Servicing Automatic Machines. Journal of Industrial Engineering. 9:28.

Papadopoulos, H. T., C. Heavey, and J. Browne. 1993. Queueing Theory in Manufacturing Systems: Analysis and Design. 1st ed. London: Chapman & Hall.

Park, S., G. T. Mackulak, and J. W. Fowler. 2001. An Overall Framework for Generating Simulation-Based Cycle Time-Throughput Curves. Proc. Winter Sim. Conf. 2: 1178–1187.

Peck, L. G. and Hazelwood, R. N. 1958. Finite Queueing Tables. New York: John Wiley & Sons.

Pollaczek, F. 1932. Losung eines Geometrischen Wahrscheinlichkeits-Problems. Math. Z. 35: 230–278.

Reynolds, G. H. 1975. An M/M/m/n Queue for the Shortest Distance Priority Machine Interference Problem. Operations Research. 23(2): 325–341.

Reiser, M. and Kobayashi, H. 1974. Accuracy of the Diffusion Approximation for Some Queueing Systems. IBM Journal of Research and Development. 18(2): 110–124.

Reiser M. and S. S. Lavenberg. 1980. Mean-Value Analysis of Closed Multichain Queuing Networks. J. ACM. 27(2): 313–322.

Resnick, S. I. 2005. Adventures in Stochastic Processes. Boston: Birkhauser.

Rose, O. 2001. The Shortest Processing Time First (SPTF) Dispatch Rule and Some Variants in Semiconductor Manufacturing. Proc. Winter Sim. Conf. 2: 1220–1224.

Ruelle, O. 1997. Continuous Flow Manufacturing: The Ultimate Theory of Constraints. Proc. IEEE /SEMI ASMC: 216–221.

Sakasegawa. H. 1977. An approximation Formula $L_q = \alpha\rho^{\beta}/(1 - \rho)$. Annual of the Institute for Statistical Mathematics. 29 (A): 67–75.

Sattler, L. 1996. Using Queueing Curve Approximations in a Fab to Determine Productivity Improvements. Proc. IEEE/SEMI ASMC: 140–145.

Schweitzer, P. 1979. Approximate Analysis of Multiclass Closed Networks of Queues. International Conference on Stochastic Control and Optimization.

Serfozo, R. 1999. Introduction to Stochastic Networks. New York: Springer.

Segal, M. and W. Whitt. 1989. A Queueing Network Analyzer for Manufacturing. Teletraffic Science for New Cost-effective Systems: 1146–1152.

SEMI E10. 2001. Specification for Definition and Measurement of Equipment Reliability, Availability, and Main-tainability, Book of SEMI Standards. Mountain View, CA: SEMI.

Shanthikumar, J. G. and J. A. Buzacott. 1980. On the Approximations to the Single Server Queue. International Journal of Production Research. 18(6): 761–773.

Shanthikumar, J. G. and J. A. Buzacott. 1981. Open Queueing Network Models of Dynamic Job Shops. International Journal of Production Research. 19(3): 255–266.

Smith, W. L. 1953. Distribution of Queueing Times. Proc. Comb. Phil. Soc. 49: 449–461.

Spearman, Mark. 1991. An Analytic Congestion Model for Closed Production Systems with IFR Processing Times. Management Science. 37(8): 1015–1029.

Stecke, K. E. and J. E. Aronson. 1985. Review of Operator/Machine Interference Models. Int. J. Prod. Res. 23(1): 129–151.

Suresh, S. and W. Whitt. 1990a. Arranging Queues in Series: A Simulation Experiment. Management Science. 36(9): 1080–1090.

Suresh, S. and W. Whitt. 1990b. The Heavy-Traffic Bottleneck Phenomenon in Open Queueing Networks. Operations Research Letters. 9: 355–362.

Suri, R., J. L. Sanders and M. Kamath. 1993. Performance Evaluation of Production Networks. Handbooks in OR & MS, 4: 199–286.

Suri, R., S. Sahu and M. Vernon. 2007. Approximate Mean Value Analysis for Closed Queueing Networks with Multiple-Server Stations. Proceedings of the 2007 Industrial Engineering Research Conference.

Tembe, S. V. and R. W. Wolff. 1974. The Optimal Order of Service in Tandem Queues. Operations Research. 24: 824–832.

Wang, K. H. and Sivazlian, B. D. 1990. Comparative Analysis for the G/G/R Machine Repair Problem. Computers & Industrial Engineering. 18(4): 511–520.

Weber, R. R. 1979. The Interchangeability of ./M/1 Queues in Series. Journal of Applied Probability. 16(3): 690–695.

White, H. and L. Christie. 1958. Queueing with Preemptive Priorities or with Breakdown. Oper. Res. 6: 79–95.

Whitt, W. 1983. The Queueing Network Analyzer. The Bell System Technical Journal. 62(9): 2779–2815.

Whitt, W. 1984. Approximations for Departure Process and Queues in Series. Naval Research Logistics Quarterly. 31: 499–521.

Whitt, W. 1985. The Best Order for Queues in Series. Management Science. 31(4): 475–487.

Whitt, W. 1993. Approximations for the GI/G/m queue. Production and Operations Management. 2(2): 114–161.

Whitt, W. 2003. Stochastic Process Limits. New York: Springer.

Wolff, R. W. 1982. Poisson Arrivals See Time Averages. Operations Research. 30: 223–231.

Wolff, R. W. 1989. Stochastic Modeling and the Theory of Queues. New Jersey: Prentice-Hall.

Wu, K. 2005. An Examination of Variability and its Basic Properties for a Factory. IEEE Trans. Semi. Manu. 18(1): 214-221.

Wu, K. and K. Hui. 2007. The determination and Indetermination of Service Times in Manufacturing Systems. IEEE Trans. Semi. Manu. 21:72–82.

Wu, K., L. F. McGinnis, and B. Zwart. 2007. Compatibility of Queueing Theory, Manufacturing Systems and SEMI Standards. Submitted to IEEE CASE.

Wu, K. and K. Hui. 2008. The Determination and Indetermination of Service Times in Manufacturing Systems. IEEE Trans. Semi. Manu. 21(1): 72–82.

Wu, K., L. F. McGinnis, and B. Zwart. 2008. Queueing Models for Single Machine Manufacturing Systems with Interruptions. Winter Sim. Conf.

Yang, F., B. Ankenman, and B. L. Nelson. 2007. Efficient Generation of Cycle Time-Throughput Curves through Simulation and Metamodeling. Naval Research Logistics. 54(1): 78–93.

VITA

KAN WU

WU received the B.S. degree in nuclear engineering from National Tsing Hua University, Hsinchu, Taiwan. He received the M.S. degree in industrial engineering and operations research and the M.E. degree in nuclear engineering from the University of California, Berkeley in 1996. Afterward, he was a consultant engineer with Tefen, Ltd., and a senior engineer with Taiwan Semiconductor Manufacturing Company. During 2003 to 2005, he was an IE manager at Inotera Memories Inc. His research interests include production planning, scheduling, and dispatching in manufacturing systems.